INTERNATIONAL STANDARD

ISO 10303-232

First edition
2002-06-01

Industrial automation systems and integration — Product data representation and exchange —

Part 232: Application protocol: Technical data packaging core information and exchange

Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits —

Partie 232: Protocole d'application: Information centrale et échange de paquetage de données techniques

Reference number
ISO 10303-232:2002(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Page

1 Scope 1

2 Normative references 4

3 Terms, definitions, and abbreviations 7
3.1 Terms defined in ISO 10303-1 7
3.2 Terms defined in ISO 10303-31 8
3.3 Terms defined in ISO 10303-42 8
3.4 Terms defined in ISO 10303-203 9
3.5 Other terms and definitions 10
3.6 Abbreviations 14

4 Information requirements 15
4.1 Units of functionality 16
4.1.1 data_definition_exchange 26
4.1.2 data_list 31
4.1.3 drawing 34
4.1.4 indentured_data_list 36
4.1.5 index_list 40
4.1.6 other_list 42
4.1.7 parts_list 45
4.1.8 presentation 49
4.1.9 product_data_set 51
4.1.10 reference_document 54
4.2 Application objects 55
4.3 Application assertions 239

5 Application interpreted model 310
5.1 Mapping table 310
5.2 AIM EXPRESS short listing 939

6 Conformance requirements 1018
6.1 Data definition exchange for files, conformance class 1 1021
6.2 Data definition exchange for TDP elements, conformance class 2 1021
6.3 Data definition exchange for indentured methods, conformance class 3 1021
6.4 Parts list, conformance class 4 1021
6.5 Data list, conformance class 5 1021
6.6 Indentured data list, conformance class 6 1022
6.7 Index list, conformance class 7 1022
6.8 Other list, conformance class 8 1022
6.9 List with presentation, conformance class 9 1022
6.10 Reference document identification and drawing identification, Conformance Class 10 . 1022
6.11 Reference document identification and drawing identification with ISO 10303-201 and ISO 10303-202 drawing presentation identification, Conformance Class 11 1023
6.12 Product data set (PDS) without presentation format, Conformance Class 12 1023
6.13 Product data set (PDS) with shading, conformance class 13 1024

6.14 Product data set (PDS) with presentation format, conformance class 14 1024

Annex A (normative) AIM EXPRESS expanded listing 1045

Annex B (normative) AIM short names of entities 1206

Annex C (normative) Implementation method specific requirements 1219

Annex D (normative) Protocol implementation conformance statement (PICS) proforma 1223

Annex E (normative) Information object registration 1225
E.1 Document identification 1225
E.2 Schema identification 1225

Annex F (informative) Application activity model 1226
F.1 Application activity model definitions 1227
F.2 Application activity model abbreviations 1248
F.3 Application activity model diagrams 1250

Annex G (informative) Application reference model 1268

Annex H (informative) AIM EXPRESS-G 1306

Annex J (informative) Computer-interpretable listings 1373

Annex K (informative) Usage Scenarios 1374
K.1 Information exchange scenarios 1374
K.2 Definitions and levels of capability for the parts list 1381
K.3 Definitions and levels of capability for the data list 1402
K.4 Definitions and levels of capability for the index list 1417
K.5 Definitions and levels of capability for the Data Definition Exchange 1429
K.6 Definitions and levels of capability for the indentured data list 1455

Bibliography 1479

Index 1480

Figures

Figure 1 — PDM information area ISO 10303-232 addresses xi
Figure 2 — High level planning model xii
Figure 3 — Functional usage of technical data 2
Figure 4 — Format style for date 70
Figure 5 — Graphics utilized in presenting a tabulated listing 179
Figure 6 — Page anchor location 180
Figure 7 — Standard cross sections 225
Figure C.1 — Exchange structure without ISO 10303-201 or ISO 10303-202 1220
Figure C.2 — Exchange structure with ISO 10303-505 as sheet 1221
Figure C.3 — Exchange structure with ISO 10303-505 as drawing 1222

Figure F.1 — A-0: develop, procure, build, use, and maintain a product 1251
Figure F.2 — A0: develop, procure, build, use, and maintain a product 1252
Figure F.3 — A2: manage, design, and support a product 1253
Figure F.4 — A22: design and analyze a product 1254
Figure F.5 — A223: perform product detail and design analysis 1255
Figure F.6 — A2232: create detail product design 1256
Figure F.7 — A22323: prepare product models and drawings 1257
Figure F.8 — A223232: create product layouts and models 1258
Figure F.9 — A223233: create product drawing data 1259
Figure F.10 — A2232332: prepare detail product item drawings 1260
Figure F.11 — A22324: build product associated lists 1261
Figure F.12 — A24: prepare product data for use 1262
Figure F.13 — A3: use and maintain a product 1263
Figure F.14 — A33: support product 1264
Figure F.15 — A331: spares procurement 1265
Figure F.16 — A223: local manufacture 1266
Figure F.17 — A333: develop modifications 1267
Figure G.1 — ARM diagram (1 of 33) 1273
Figure G.2 — ARM diagram (2 of 33) 1274
Figure G.3 — ARM diagram (3 of 33) 1275
Figure G.4 — ARM diagram (4 of 33) 1276
Figure G.5 — ARM diagram (5 of 33) 1277
Figure G.6 — ARM diagram (6 of 33) 1278
Figure G.7 — ARM diagram (7 of 33) 1279
Figure G.8 — ARM diagram (8 of 33) 1280
Figure G.9 — ARM diagram (9 of 33) 1281
Figure G.10 — ARM diagram (10 of 33) 1282
Figure G.11 — ARM diagram (11 of 33) 1283
Figure G.12 — ARM diagram (12 of 33) 1284
Figure G.13 — ARM diagram (13 of 33) 1285
Figure G.14 — ARM diagram (14 of 33) 1286
Figure G.15 — ARM diagram (15 of 33) 1287
Figure G.16 — ARM diagram (16 of 33) 1288
Figure G.17 — ARM diagram (17 of 33) 1289
Figure G.18 — ARM diagram (18 of 33) 1290
Figure G.19 — ARM diagram (19 of 33) 1291
Figure G.20 — ARM diagram (20 of 33) 1292
Figure G.21 — ARM diagram (21 of 33) 1293
Figure G.22 — ARM diagram (22 of 33) 1294
Figure G.23 — ARM diagram (23 of 33) 1295
Figure G.24 — ARM diagram (24 of 33) 1296
Figure G.25 — ARM diagram (25 of 33) 1297
Figure G.26 — ARM diagram (26 of 33) 1298
Figure G.27 — ARM diagram (27 of 33) 1299
Figure G.28 — ARM diagram (28 of 33) 1300
Figure G.29 — ARM diagram (29 of 33) 1301
Figure G.30 — ARM diagram (30 of 33) 1302
Figure G.31 — ARM diagram (31 of 33) 1303
Figure G.32 — ARM diagram (32 of 33) 1304

Figure G.33 — ARM diagram (33 of 33) 1305
Figure H.1 — AIM EXPRESS-G diagram (1 of 66) 1307
Figure H.2 — AIM EXPRESS-G diagram (2 of 66) 1308
Figure H.3 — AIM EXPRESS-G diagram (3 of 66) 1309
Figure H.4 — AIM EXPRESS-G diagram (4 of 66) 1310
Figure H.5 — AIM EXPRESS-G diagram (5 of 66) 1311
Figure H.6 — AIM EXPRESS-G diagram (6 of 66) 1312
Figure H.7 — AIM EXPRESS-G diagram (7 of 66) 1313
Figure H.8 — AIM EXPRESS-G diagram (8 of 66) 1314
Figure H.9 — AIM EXPRESS-G diagram (9 of 66) 1315
Figure H.10 — AIM EXPRESS-G diagram (10 of 66) 1316
Figure H.11 — AIM EXPRESS-G diagram (11 of 66) 1317
Figure H.12 — AIM EXPRESS-G diagram (12 of 66) 1318
Figure H.13 — AIM EXPRESS-G diagram (13 of 66) 1319
Figure H.14 — AIM EXPRESS-G diagram (14 of 66) 1320
Figure H.15 — AIM EXPRESS-G diagram (15 of 66) 1321
Figure H.16 — AIM EXPRESS-G diagram (16 of 66) 1322
Figure H.17 — AIM EXPRESS-G diagram (17 of 66) 1323
Figure H.18 — AIM EXPRESS-G diagram (18 of 66) 1324
Figure H.19 — AIM EXPRESS-G diagram (19 of 66) 1325
Figure H.20 — AIM EXPRESS-G diagram (20 of 66) 1326
Figure H.21 — AIM EXPRESS-G diagram (21 of 66) 1327
Figure H.22 — AIM EXPRESS-G diagram (22 of 66) 1328
Figure H.23 — AIM EXPRESS-G diagram (23 of 66) 1329
Figure H.24 — AIM EXPRESS-G diagram (24 of 66) 1330
Figure H.25 — AIM EXPRESS-G diagram (25 of 66) 1331
Figure H.26 — AIM EXPRESS-G diagram (26 of 66) 1332
Figure H.27 — AIM EXPRESS-G diagram (27 of 66) 1333
Figure H.28 — AIM EXPRESS-G diagram (28 of 66) 1334
Figure H.29 — AIM EXPRESS-G diagram (29 of 66) 1335
Figure H.30 — AIM EXPRESS-G diagram (30 of 66) 1336
Figure H.31 — AIM EXPRESS-G diagram (31 of 66) 1337
Figure H.32 — AIM EXPRESS-G diagram (32 of 66) 1338
Figure H.33 — AIM EXPRESS-G diagram (33 of 66) 1339
Figure H.34 — AIM EXPRESS-G diagram (34 of 66) 1340
Figure H.35 — AIM EXPRESS-G diagram (35 of 66) 1341
Figure H.36 — AIM EXPRESS-G diagram (36 of 66) 1342
Figure H.37 — AIM EXPRESS-G diagram (37 of 66) 1343
Figure H.38 — AIM EXPRESS-G diagram (38 of 66) 1344
Figure H.39 — AIM EXPRESS-G diagram (39 of 66) 1345
Figure H.40 — AIM EXPRESS-G diagram (40 of 66) 1346
Figure H.41 — AIM EXPRESS-G diagram (41 of 66) 1347
Figure H.42 — AIM EXPRESS-G diagram (42 of 66) 1348
Figure H.43 — AIM EXPRESS-G diagram (43 of 66) 1349
Figure H.44 — AIM EXPRESS-G diagram (44 of 66) 1350
Figure H.45 — AIM EXPRESS-G diagram (45 of 66) 1351
Figure H.46 — AIM EXPRESS-G diagram (46 of 66) 1352
Figure H.47 — AIM EXPRESS-G diagram (47 of 66) 1353
Figure H.48 — AIM EXPRESS-G diagram (48 of 66) 1354

Figure H.49 — AIM EXPRESS-G diagram (49 of 66) 1355
Figure H.50 — AIM EXPRESS-G diagram (50 of 66) 1356
Figure H.51 — AIM EXPRESS-G diagram (51 of 66) 1357
Figure H.52 — AIM EXPRESS-G diagram (52 of 66) 1358
Figure H.53 — AIM EXPRESS-G diagram (53 of 66) 1359
Figure H.54 — AIM EXPRESS-G diagram (54 of 66) 1360
Figure H.55 — AIM EXPRESS-G diagram (55 of 66) 1361
Figure H.56 — AIM EXPRESS-G diagram (56 of 66) 1362
Figure H.57 — AIM EXPRESS-G diagram (57 of 66) 1363
Figure H.58 — AIM EXPRESS-G diagram (58 of 66) 1364
Figure H.59 — AIM EXPRESS-G diagram (59 of 66) 1365
Figure H.60 — AIM EXPRESS-G diagram (60 of 66) 1366
Figure H.61 — AIM EXPRESS-G diagram (61 of 66) 1367
Figure H.62 — AIM EXPRESS-G diagram (62 of 66) 1368
Figure H.63 — AIM EXPRESS-G diagram (63 of 66) 1369
Figure H.64 — AIM EXPRESS-G diagram (64 of 66) 1370
Figure H.65 — AIM EXPRESS-G diagram (65 of 66) 1371
Figure H.66 — AIM EXPRESS-G diagram (66 of 66) 1372
Figure K.1 — TDP information exchange activity 1375
Figure K.2 — Information creation scenario for TDP development 1378

Tables

Table 1 — Applicability of selects and enumerations to units of functionality 17
Table 2 — Applicability of types and entities to the units of functionality 18
Table 3 — Change_identification attribute relationships 68
Table 4 — Mapping table for common 312
Table 5 — Mapping table for data_definition_exchange 389
Table 6 — Mapping table for data_list 519
Table 7 — Maping table for drawing 566
Table 8 — Mapping table for indentured_data_list 594
Table 9 — Mapping table for index_list 694
Table 10 — Mapping table for other_list 719
Table 11 — Mapping table for parts_list 749
Table 12 — Mapping table for presentation 853
Table 13 — Mapping table for product_data_set 874
Table 14 — Mapping table for reference_document 931
Table 15 — Conformance classes 1019
Table 16 — Conformance class elements 1024
Table K.2.1 — Levels of capability for PL selects and enumerations 1383
Table K.2.2 — Levels of capability for PL application objects and attributes 1384
Table K.3.1 — Levels of capability for DL selects and enumerations 1404
Table K.3.2 — Levels of capability for DL application objects and attributes 1405
Table K.4.1 — Levels of capability for IL selects and enumerations 1419
Table K.4.2 — Levels of capability for IL application objects and attributes 1419
Table K.5.1 — Parent child relationships used in indenture list types 1432
Table K.5.2 — Levels of capability for DDE, selects, and enumerations 1435
Table K.5.3 — Levels of capability for DDE application objects and attributes 1438
Table K.6.1 — Parent child relationships used in indenture list types 1458

Table K.6.2 — Levels of capability for IDL selects, and enumerations 1461
Table K.6.3 — Levels of capability for IDL application objects and attributes 1463

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 10303 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10303-232 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

This International Standard is organized as a series of parts, each published separately. The structure of this standard is described in ISO 10303-1.

Each part of this International Standard is a member of one of the following series: description methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part is a member of the application protocols series.

A complete list of parts of ISO 10303 is available from the Internet:

<http://www.nist.gov/sc4/editing/step/titles/>

Should further parts of ISO 10303 be published, they will follow the same numbering pattern.

Annexes A, B, C, D and E form a normative part of this part of ISO 10303. Annexes F, G, H, J and K are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their lifecycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and as a basis for archiving.

This part of ISO 10303 is a member of the application protocol series. This part of ISO 10303 specifies an application protocol (AP) for packaging product data so that configuration controlled access and data exchanges can be achieved among Product Data Management (PDM) systems.

This part of ISO 10303 identifies specific groupings of product information for the access or exchange of data, such as part lists and drawings. An integrated packing list of these product information groups can be captured by this part of ISO 10303 which satisfies data accountability requirements during data access or a data exchange. The actual format of the data being accessed or exchanged may be through ISO 10303 applications protocols (including this part of ISO 10303), through other ISO standards that define computer file formats, or any other computer file format that is agreed to by parties involved in the data access or data exchange agreement. This capability will satisfy the industrial need to communicate and share the total design definition of a product from an overall product data configuration structure perspective among organizations, partners, vendors, and customers. Portions of product data can be exchanged with the knowledge of where that portion belongs in the overall product data configuration structure. The goal of this part of ISO 10303 is to provide an information structure wherein product information can be electronically captured and managed from both a product information item perspective and a document based perspective, such as in PDM systems.

Current PDM systems being installed in industry manage documents from a product view point. No ISO 10303 Application Protocol addresses a document configuration for product data structures. This part of ISO 10303 addresses this document configuration structure void. Enterprises and industries that have large infrastructures that rely on document based management systems will utilize this part of ISO 10303 to migrate from their current implementations into a configuration control of product data from a product item perspective.

There are two aspects to this part of ISO 10303. The first is the packaging of product data for exchange or access. The second is to provide the exchange requirements of individual product data groupings focusing on associated list information such as data list, index list, indentured data list, and parts list.

The packaging aspect provides the requirements for collecting, organizing, and managing the exchange of a complex set of data files or database views representing the different product data groupings that identify and define a product. These packaging aspects reside in a PDM's underlying metadata. A product data grouping defines a particular view of product information and may be identified and managed as a document, or a product data set or a unique view within a database. Drawings, Associated Lists, and Reference Documents are considered product data groupings. The practice of packaging product data into groupings is called an Engineering Data Package or a Technical Data Package (TDP). As a result of this packaging, this part of ISO 10303 defines the interoperation of other parts of ISO 10303 (for example, ISO 10303-203 and ISO 10303-202) and the managed inclusion of a mixed set of standards for representation of the various TDP elements, such as simple product data groupings.

Figure 1 is the high level functions of a PDM system. The areas enclosed in Figure 1 identify the information this part of ISO 10303 will support.

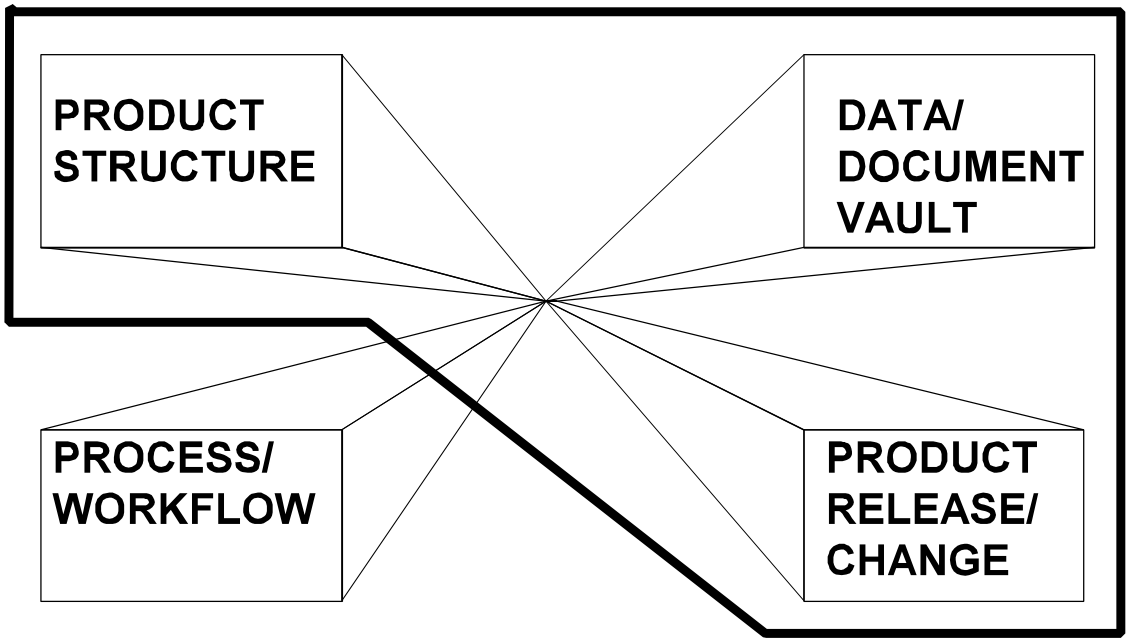

Figure 1 — PDM information area ISO 10303-232 addresses

Figure 2 is a high level planning model that describes some of the basic information concepts and their relationships contained in this part of ISO 10303. These concepts and relationships include the following:

— identification of a product and the relationship of that product to its constituents;

— identification of the TDP elements and the relationship of those TDP elements to each other;

— identification of the relationship between a product and its TDP element or the relationship among products and their TDP elements;

— the documentation of formal change level and release status of product information and the TDP elements for the product information;

— the definition and presentation of associated list information;

— the packaging of the TDP elements for access or exchange.

NOTE 1 Annex K provides comprehensive descriptions for many of the capabilities this part of ISO 10303 provides.

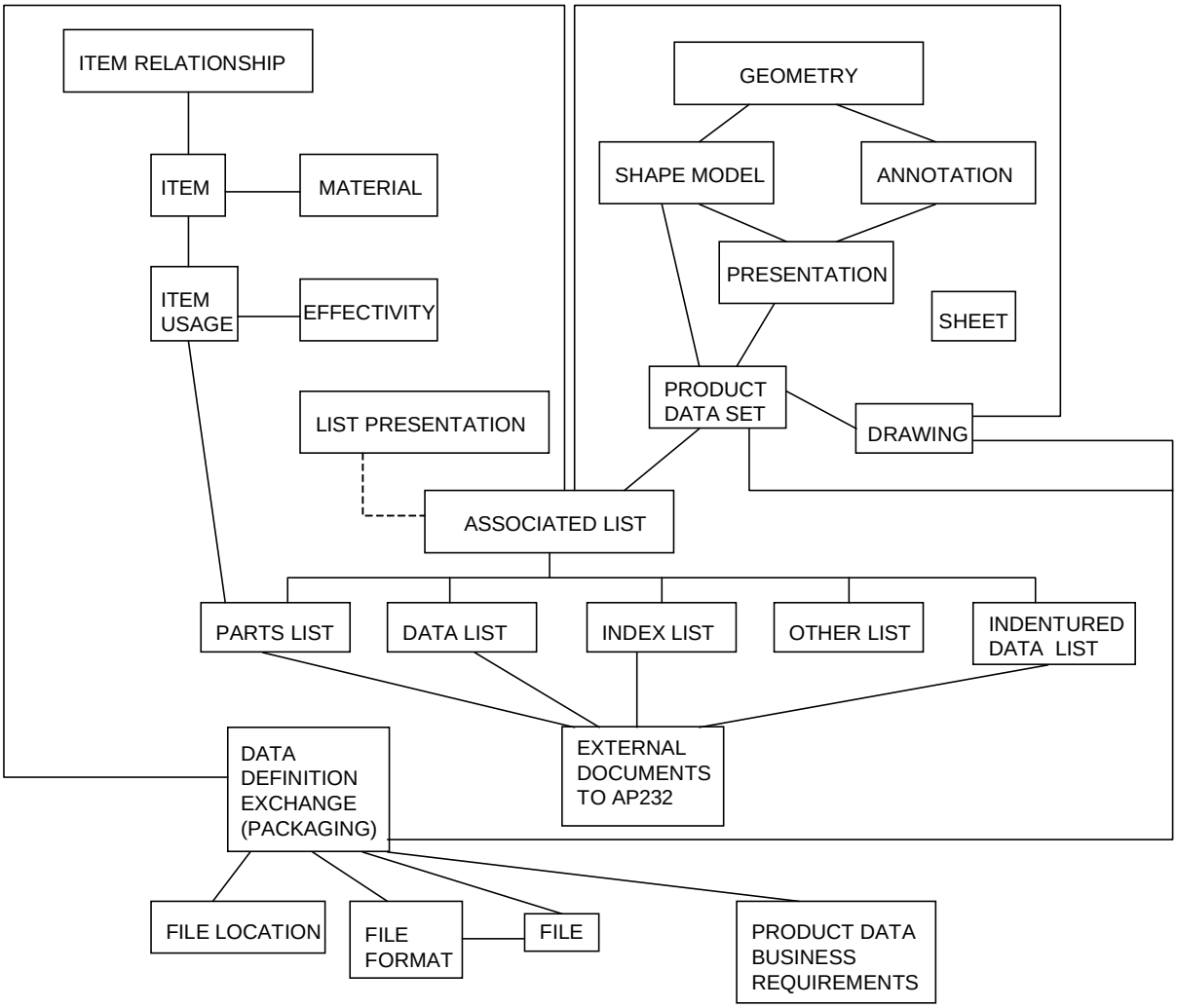

Figure 2 — High level planning model

This application protocol defines the context, scope, and information requirements for the content and exchange of technical data packages and specifies the integrated resources necessary to satisfy these requirements.

NOTE 2 Technical data packages may define products, component parts, assemblies, and their related documentation. TDPs are not confined to mechanical definitions of a product, but may define TDPs that define functionality of a product, TDPs that define concepts for a product, or TDPs that define form and fit of a product.

NOTE 3 This part of ISO 10303 may define interoperation of other ISO 10303 APs within a TDP.

NOTE 4 This part of ISO 10303 may identify the co-existence of ISO 10303 APs with different data representations for the same product information.

NOTE 5 This part of ISO 10303 may identify product data stored in other computer data file formats and the relationship of the data contained in the file with other product data contained in other computer data files.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirements of the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in annex H. Additional requirements for specific implementation methods are given in annex C.

NOTE 6 ISO 10303 standards are referenced throughout the documentation of the requirements of this part of ISO 10303. This part utilizes ISO 10303 standards through the use of application interpreted constructs.

Industrial automation systems and integration — Product data representation and exchange —

Part 232: Application protocol: Technical data packaging core information and exchange

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for Technical Data Packages (TDPs) (see 3.5.28) to be exchanged among product data management systems.

Each enterprise uses content, format, and the level of configuration control as parameters when establishing its product exchange or access requirements among business partners. Because of the diverse set of products, product data, and lifecycle processes PDM systems support, this part of ISO 10303 allows many combinations of these parameters and their values.

Using a defined set of these parameters, the disclosure of product information needs to be sufficient to satisfy the business purpose of the TDP.

NOTE 1 TDPs may be prepared to a level where the product information is sufficient to evaluate a product definition concept. Or a TDP may be prepared to a level where the product information is sufficient to enable full design disclosure (see 3.5.9).

Requirements for this part of ISO 10303 were derived from functions that create and use TDPs and reside throughout the product's life cycle. The key informational aspects addressed in this part of ISO 10303 are shared and exchanged throughout the product's life cycle.

NOTE 2 Within a product's life cycle, there are many functions that create and use the technical information about a product. Figure 3 illustrates the functional usage of technical data within each life cycle phase of a product. The largest percentage of the technical data is developed in the concept development, concept and validation, and product and process development lifecycle phases. The operations and support lifecycle phase, for most products or commodities, is the longest and is impacted the greatest by the quality and usability of the TDP information. The production lifecycle phase typically has the second largest usage of the information contained within the TDP. In the production and product process development lifecycle phase, TDP data is used to build and deliver the product.

NOTE 3 The application activity model in annex F provides a graphical representation of the processes and information flows that are the basis for the definition of the scope of this part of ISO 10303.

EXAMPLE 1 The following represent different types of Technical Data Packages through a product's life cycle:

— Conceptual Design Drawings and Associated Lists;

— Developmental Design Drawings and Associated Lists;

— Product Drawings and Associated Lists;

— Commercial Drawings and Associated Lists;

— Special Inspection Equipment Drawings and Associated Lists;

— Special Tooling Drawings and Associated Lists.

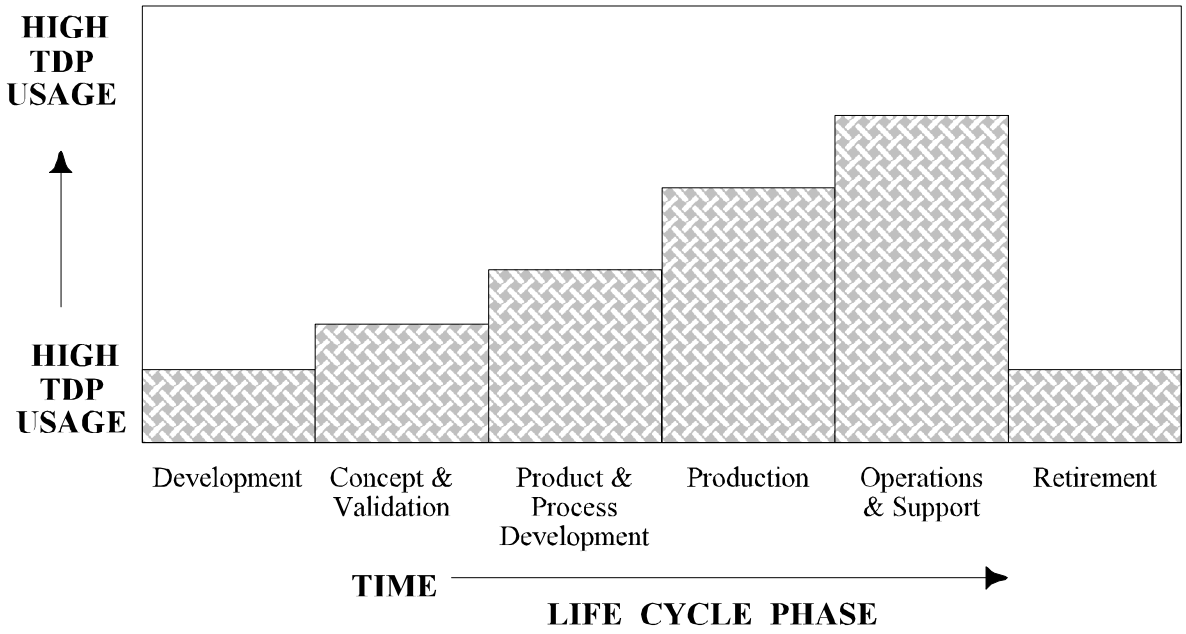

Figure 3 — Functional usage of technical data

NOTE 4 Definitions for the preceding types of TDPs are given in Clause 3.5.

The content, form, and format of the data are critical to the core information content for exchange or access of the TDP. The following are within the scope of this part of ISO 10303:

— All products and their commodity types;

NOTE 5 This part of ISO 10303 is defined independently of product or commodity.

— Product definition data and product configuration control data pertaining to the concept development, concept and validation, product and process development, production, operations and support, and retirement phases of a product (herein called the product life cycle);

NOTE 6 ISO 10303 standards are referenced throughout the documentation of the requirements of this part of ISO 10303. This part utilizes ISO 10303 standards through the use of application interpreted constructs.

— Relationship of the product to a technical data package element (see 3.5.27);

— Identification of drawings related to the product that require configuration control, exchange, or access;

— The data content requirements for parts lists, data lists, index lists, indentured data lists, and other associated lists that are associated to a drawing or a product data set (see 3.5 for definitions);

— The data content requirements for product data set that represent geometric product shape;

— Identification of files containing shape information and the positioning information needed to place components in an assembly;

— Identification of alternate geometric representations of the product definition data by different disciplines during the product life cycle;

— Identification of any group of technical data related to the product that needs configuration control exchange or access;

EXAMPLE 2 A finite element analysis (models, controls, and results) and testing reports are each a group of technical data.

— Identification of specifications and standards that define or describe the product or product unique processes;

— Identification of documentation that define or describe change activity to the product or product related documentation;

EXAMPLE 3 Change activity is documented in the form of Drawing Revision Notices, Engineering Change Notices, and Drawing Change Notices.

— The Identification of standard parts (see 3.5.26) for the purpose of their inclusion in a product's design;

— The visual presentation for human understanding of the associated list data and the product data set;

— The data requirements for configuration control exchange of Technical Data Packages;

NOTE 7 Identification of document version will be defined by this part of ISO 10303.

— The identification and relationship of Technical Data Package elements within a TDP exchange;

— The identification of file and file format for Technical Data Package elements;

NOTE 8 File and file format information may be defined by this part of ISO 10303, another part of ISO 10303, National Standards or through mutual agreement between the sending and receiving parties of the TDP elements.

— The identification of digital and non-digital media for specific technical data package elements.

EXAMPLE 4 Digital media could include floppy disk, diskette, compact disk, and 9-track tape.

EXAMPLE 5 Non-digital media could include paper, mylar, aperture card, and vellum.

The following are outside the scope of this part of ISO 10303:

— Data content requirements for technological data used in, or resulting from, the analysis or test of a design that is used as evidence for consideration of a change to a design;

NOTE 9 A product data set may contain analysis or test data in a format other than this part of ISO 10303.

— Data content requirements for technological data that results in changes to the design during the initial design evolution prior to release;

NOTE 10 Changes prior to release are considered informal change activity.

NOTE 11 ISO 10303-209 [4] defines requirements for informal change activity prior to engineering release.

— Data content requirements for business management data for a design project;

EXAMPLE 6 Business data could include schedule, cost, time standards, risk, and related management information.

— Data content requirements for alternate representations of the data by different disciplines outside of that required to define, manufacture, or procure the product;

— Data content requirements for definition of digital or non-digital media for a TDP exchange;

— Data content requirements for definition of procedures to record the digital TDP files to digital media.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303-232. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303-232 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 31 (all parts), *Quantities and units*

ISO 1000:1992, *SI units and recommendations for the use of their multiples and of certain other units*

ISO/IEC 8824-1:1998, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*

ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

ISO 10303-21:2002, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*

ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts*

ISO 10303-41:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*

ISO 10303-42:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

ISO 10303-43:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures*

ISO 10303-44:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resource: Product structure configuration*

ISO 10303-45:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resource: Materials*

ISO 10303-46:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 46: Integrated generic resources: Visual presentation*

ISO 10303-49:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 49: Integrated generic resources: Processs structure and properties*

ISO 10303-101:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 101: Integrated application resources: Draughting*

ISO 10303-201:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 201: Application protocol: Explicit draughting*

ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Application protocol: Associative draughting*

ISO 10303-203:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol: Configuration controlled design*

ISO 10303-501:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 501: Application interpreted construct: Edge-based wireframe*

ISO 10303-502:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 502: Application interpreted construct: Shell-based wireframe*

ISO 10303-503:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 503: Application interpreted construct: Geometrically bounded 2D wireframe*

ISO 10303-505:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 505: Application interpreted construct: Drawing structure and administration*

ISO 10303-507:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 507: Application interpreted construct: Geometrically bounded surface*

ISO 10303-508:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 508: Application interpreted construct: Non-manifold surface*

ISO 10303-509:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 509: Application interpreted construct: Manifold surface*

ISO 10303-510:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 510: Application interpreted construct: Geometrically bounded wireframe*

ISO 10303-511:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 511: Application interpreted construct: Topologically bounded surface*

ISO 10303-512:1999, *Industrial automation systems and integration — Product data representation and exchange — Part 512: Application interpreted construct: Faceted boundary representation*

ISO 10303-513:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 513: Application interpreted construct: Elementary boundary representation*

ISO 10303-514:1999, *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation*

ISO 10303-515:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 515: Application interpreted construct: Constructive solid geometry*

ISO 10303-517:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 517: Application interpreted construct: Mechanical design geometric presentation*

ISO 10303-518—[1)], *Industrial automation systems and integration — Product data representation and exchange — Part 518: Application interpreted construct: Mechanical design shaded presentation*

[1)] To be published.

3 Terms, definitions, and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-1 apply:

— abstract test suite;

— application;

— application activity model;

— application context;

— application interpreted model;

— application object;

— application protocol;

— application reference model;

— assembly;

— component;

— conformance class;

— conformance requirement;

— data;

— data exchange;

— implementation method;

— information;

— integrated resource;

— interpretation;

— PICS proforma;

— product;

— product data;

— protocol implementation conformance statement;

— structure;

— unit of functionality.

3.2 Terms defined in ISO 10303-31

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-31 apply:

— conformance testing;

— preprocessor;

— postprocessor.

3.3 Terms defined in ISO 10303-42

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-42 apply:

— arcwise connected;

— axi-symmetric;

— bounds;

— boundary;

— boundary representation solid model;

— closed curve;

— closed surface;

— connected;

— connected component;

— curve;

— cycle;

— dimensionality;

— *d*-manifold with boundary;

— domain;

— extent;

— finite;

— geometric coordinate system;

— graph;

— handle;

— homomorphic;

— list;

— open curve;

— open surface;

— orientable;

— overlap;

— parameter range;

— parameter space;

— placement coordinate system;

— self-intersect;

— self-loop;

— set;

— space dimensionality;

— surface;

— topological sense.

3.4 Terms defined in ISO 10303-203

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-203 apply:

— boundary representation model;

— design phase;

— mechanical part;

— solid model;

— sub-assembly;

— wireframe model.

3.5 Other terms and definitions

For the purposes of this part of ISO 10303, the following terms and definitions apply:

3.5.1
associated list
a tabulation of pertinent engineering information pertaining to an item depicted on an engineering drawing, on a set of engineering drawings, a product data set, in a product data set, or a combination thereof

EXAMPLE Examples of an associated list are Parts List, Data List, Indentured Data List, and Index List.

3.5.2
commercial drawings
engineering and design disclosure information in support of end items that are commercially developed items, or off-the-shelf items

3.5.3
computer interpretable data
data that is in a specific context and format and can be read and stored in a computer system such that action may be taken based on the content of the data

EXAMPLE A properly formatted floppy disk contains a file in a standard file format. One of the parameters of the data in the file is release information, which is one of the following: "released", "not released", or "unknown". If the release information for this file is "not released". The software application reading this file places this information in a working area, rather than the release area.

3.5.4
computer processable data
data that is in a format that can be read and stored by a computer

EXAMPLE A personal note written in a commercial word processor format on a properly formatted floppy disk can be read by a personal computer (PC) with the appropriate software. The data can also be displayed and stored on that PC according to the formatting instructions, regardless of the note content. However, a hand written note is not computer processable.

3.5.5
conceptual design drawings
design concepts in defined graphic form, and including appropriate textual information required for analysis and evaluation of those concepts

3.5.6
data definition exchange (DDE)
a listing that identifies and records the total number of TDP elements required to support the given product data for an item, assembly, or detail being exchanged or available for access

3.5.7
data list
a tabulation of all engineering drawings, associated lists, specifications, standards, product data sets and subordinate data lists pertaining to the item that applies to the data list

3.5.8
design activity standard
a document containing a set of rules, criteria, and requirements established by an enterprise, organization, or department used to measure the accuracy and control completeness of a design

NOTE 1 A standard may be developed by the organization that is doing the design.

NOTE 2 A standard may be developed by an industrial organization (for example, ISO) that is concerned with standards development. A design activity may adopt the standard without modification or the organization may modify the standard for their purpose.

3.5.9
design disclosure
engineering definition sufficiently complete to enable a competent manufacturer to produce and maintain quality control of item(s) to the degree that physical and performance characteristics interchangeable with those of the original design are obtained without resorting to additional product design effort, additional design data, or recourse to the original design activity

3.5.10
developmental design drawings
data provided to support the analysis of a specific design approach and the fabrication of product prototype for test or experimentation

3.5.11
drawing types
one of two types: industry standard and government peculiar, and can be typed according to the format of the data.

3.5.12
government peculiar drawings
drawings whose specific data content is defined by a specific Government program or Government entity

EXAMPLE Different government peculiar drawing types: inseparable assembly drawing, envelope drawing, and package content drawing.

3.5.13
in-house documents
documents that are owned and controlled within an enterprise

3.5.14
indentured data list
a structured tabulation of all engineering drawings, associated lists, specifications, standards, product data sets, and subordinate data lists pertaining to the item to which the data list applies and essential in-house documents (see 3.5.13) necessary to meet the technical design disclosure requirements of a project

NOTE The indentured data list can be viewed as a tabulation listed in a top-down breakdown (generation tree) order using parts or documents as its top-down breakdown order.

3.5.15
index list
a tabulation of data lists and possible subordinate index lists pertaining to a business requirement

3.5.16
industry standard drawing
drawing types that are recognized by a standards body such as ISO

EXAMPLE Different industry standard drawing types are detailed drawing, assembly drawing, and installation drawing.

3.5.17
intelligent data format
a format that captures product data in more than a presentation perspective

EXAMPLE Standard exchange for product model data.

3.5.18
parts list
a tabulation of identified parts, bulk materials, and possible reference documents used in an Item

NOTE Other terms used to describe a parts list are the following:

— list of materials;

— bill of materials;

— stock list;

— item list.

3.5.19
path information
a computer interpretable or human interpretable string of text that allows a software application or person enough knowledge to follow a trail over a computer network

3.5.20
product data set
a computer interpretable file of Product model data

EXAMPLE 3D models, and configuration data such as release, security, and/or contract information.

3.5.21
product drawings and associated lists
the necessary design, engineering, manufacturing, and quality assurance requirements information necessary to enable the procurement or manufacture of an interchangeable item that duplicates the physical and performance characteristics of the original product, without additional design engineering effort or recourse to the original design activity

3.5.22
product information
recorded facts, concepts, or instructions about a product

3.5.23
reference document
any design activity standard (see 3.5.8), drawing, or other document that is specified on a drawing or a list.

3.5.24
special inspection equipment drawings and associated lists
the data required to permit the limited production of special inspection equipment that duplicates the physical and performance characteristics of any previously built special inspection equipment

3.5.25
special tooling drawings and associated lists
the data necessary to enable a manufacturer to produce an item that duplicates the physicial and performance characteristics identical to those of the original tooling

3.5.26
standard parts
parts that an enterprise identifies as being utilized multiple times within the enterprise's business domain

3.5.27
TDP element
a piece of product information that would be exchanged or shared between two parties

EXAMPLE Such as engineering drawings, associated lists, referenced documents, or other data.

3.5.28
Technical Data Package
a collection of related information about a product being exchanged among PDM systems

EXAMPLE The following examples of TDP types identify requirements for TDP elements:

— Conceptual design drawings and associated lists;

— Developmental design drawings and associated lists;

— Product drawings and associated lists;

— Commercial drawings and associated lists;

— Specifications;

— Special inspection equipment and associated lists;

— Special tooling drawings and associated lists;

— Software and software documentation.

3.6 Abbreviations

For the purposes of this part of ISO 10303, the following abbreviations apply:

AAM application activity model

AIC application interpreted construct

AIM application interpreted model

AP application protocol

ARM application reference model

BOM bill-of-material

CAD computer aided design

CDRL contract data requirements list

CID commercial item description

CM configuration management

DAG directed acyclic graph

DDE Data Definition Exchange

DID data item description

DL Data List

DWG drawing

ICAM integrated computer-aided manufacturing

ID identification

IDEF0 ICAM definition language 0

IDL Indentured Data List

IGES Initial Graphics Exchange Specification

IL Index List

OL Other List

PDM Product Data Management

PDS Product Data Set

PICS Protocol information and conformance statement

PL Parts List

PRES Presentation

RD Reference Document

TDP Technical Data Package

UoF Units of Functionality

4 Information requirements

This clause specifies the information required for technical data packaging core information and exchange.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTE 1 A graphical representation of the information requirements is given in annex G.

NOTE 2 The information requirements correspond to those of the activities identified as being in the scope of this application protocol are in annex F.

NOTE 3 The mapping table is specified in 5.1 that shows how the information requirements are met using the integrated resources of this International Standard. The use of the integrated resources introduces additional requirements that are common to application protocols.

NOTE 4 The information requirements were captured using the EXPRESS language defined in ISO 10303-11. The EXPRESS language construces of entity, type, select, and enumeration are used to define an application object. A unit of functionality consists of a subset of application objects.

The following are high-level information requirements:

— any exchange requiring exchange document accountability shall contain at least one Data Definition Exchange (DDE) element;

— the DDE shall reference TDP elements for the information of interest;

— the TDP elements are drawings, product data sets, reference documents, associated lists, and any other piece of product information that would be exchanged or shared between two parties.

4.1 Units of functionality

This subclause specifies the units of functionality for the technical data packaging core information and exchange application protocol. This part of ISO 10303 specifies the following units of functionality:

— data_definition_exchange;

— data_list;

— drawing;

— indentured_data_list;

— index_list;

— other_list;

— parts_list;

— presentation;

— product_data_set;

— reference_document.

NOTE 1 The following ISO 10303-203 UoFs are considered to be a portion of this part of ISO 10303: authorization, bill_of_material, design_information, effectivity, end_item_identification, part_identification, and source_control

The units of functionality and a description of the functions that each UoF supports are given in 4.1.1 through 4.1.11. Table 1 and Table 2 provide the applicability of the EXPRESS constructs, selects, enumerations, types, and entities, to the units of functionality. The application objects included in the UoFs are defined in 4.2. The application objects that are used by each UoF are given in Table 2.

NOTE 2 The objects in Table 1 reflect the selects and enumerations from the ARM EXPRESS-G diagram in annex G. The objects in Table 2 reflect entities and types from the ARM EXPRESS-G. Clauses 4.1.1 thru 4.1.11 reflect the same object list as in Table 2 with the same order. Clause 4.2 incorporates all types, selects, and enumerations into the using entity's definition resulting in more robust objects.

NOTE 3 Abbreviations used in the column headings for Table 1 and Table 2 are found in 3.6.

Table 1 — Applicability of selects and enumerations to units of functionality

Objects	Units of functionality									
Select, Enumeration	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
alternate_identification_element_-select	X	X	X	X	X	X	X	X		X
alternate_identification_item_-select	X	X	X	X	X	X	X	X		X
change_type	X	X	X	X	X	X	X	X		X
component_list_type_enumeration	X									
component_list_type_select	X									
data_definition_entry_and_file_-select					X					
data_definition_entry_select					X					
data_definition_exchange_list_-method					X					
data_definition_file_entry					X					
date_or_event	X			X	X					
drawing_or_product_data_set	X	X	X	X		X				X
entry_format_select					X					
exchange_reason					X					
geometry_select	X									
graphics_parameters									X	
identification_select				X	X					
indentured_list_method				X	X					
item_identifying_number_select	X	X		X	X	X	X			
item_list_applies_to_select	X									
item_or_element				X	X					
list_entry_select				X	X					
quantity_accuracy_enumeration	X			X	X			X		

Table 1 — Applicability of selects and enumerations to units of functionality (concluded)

Objects	Units of functionality									
Select, Enumeration	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
quantity_accuracy_select	X			X	X			X		
quantity_type_select	X			X	X					
retrofit_state	X									
revision_authorization_select	X	X	X	X	X	X	X	X		X
shape_select	X			X	X	X	X			
source_identification		X		X	X					
stock_size_classification	X									
stock_size_cross_section	X									
usage_condition	X									
usage_context	X			X	X					
valid_tdp_elements_for_data_list		X								
weight_derivation	X									

Table 2 — Applicability of types and entities to the units of functionality

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
a_number	X	X	X	X	X	X	X	X	X	X
a_real	X	X	X	X	X	X	X	X	X	X
accessed_file					X					
alternate_element_identification	X	X	X	X	X	X	X	X		X
alternate_identification_of_item	X	X		X	X	X	X			
alternate_item	X			X	X					
an_integer	X	X	X	X	X	X	X	X	X	X

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
annotation						X	X			
approval	X	X	X	X	X	X	X			X
assembly_relationship	X									
associated_list	X	X	X	X						X
body_presentation									X	
certification	X	X	X	X	X	X	X	X		X
change_identification	X	X	X	X	X	X	X	X		X
column_header									X	
columnar_data_content_holder									X	
company	X	X	X	X	X	X	X	X		X
company_code	X	X	X	X	X	X	X	X		X
configuration	X	X	X	X	X	X	X			X
content_property				X	X	X				
contract	X	X	X	X	X	X	X			X
contract_submission		X		X	X					
data_definition_entry_item					X					
data_definition_entry_tdp_-element					X					
data_definition_exchange					X					
data_definition_exchange_body					X					
data_definition_exchange_header					X					
data_definition_exchange_-simple_entry					X					
data_definition_indentured_entry					X					
data_definition_indentured_list_method					X					

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
data_list		X	X							
data_list_body		X								
data_list_entry		X								
data_list_header		X	X							
data_list_tabulation		X								
date	X	X	X	X	X	X	X	X	X	X
date_effectivity	X			X	X					
delivery_accounting					X					
design_authority	X	X	X	X	X	X	X	X		X
distribution_notice	X	X	X	X	X	X	X			X
document_list		X	X	X						
document_usage_parameter	X	X	X	X	X	X	X	X		X
drawing	X	X	X	X		X	X			X
drawing_suffix_number_-combination	X	X		X	X	X	X			
effectivity	X			X	X					
element_identification	X	X	X	X	X	X	X	X		X
element_type	X	X	X	X	X	X	X	X		X
event	X			X	X					
exchange_file					X					
explicit_graphics									X	
external_graphics_file									X	
external_library_reference	X			X	X	. X				
field_content_holder									X	
field_content_holder_singular									X	

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
field_content_holder_tabulation									X	
field_location									X	
field_location_definition									X	
field_title									X	
file	X				X	X			X	
file_format	X				X	X			X	
file_relationship	X				X	X			X	
foreign_defined_item	X									
geometric_validation_property						X				
geometry						X	X		X	
general_item_definition_-relationship	X									
group_parent_column_header									X	
group_parent_content_holder									X	
group_parent_header									X	
header	X	X	X	X	X	X	X			X
header_block									X	
header_configuration_with_-element_identification	X	X	X	X	X	X	X			X
header_presentation									X	
identifier	X	X	X	X	X	X	X	X	X	X
indentured_data_list			X	X						
indentured_data_list_body				X						
indentured_data_list_entry				X						
indentured_data_list_header			X	X						

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
indentured_data_list_tabulation				X						
indentured_list_by_document				X	X					
indentured_list_by_part				X	X					
indentured_list_by_part_with_-document_references_to_parts				X	X					
independent_property	X			X	X	X				
independent_property_relationship	X			X	X	X				
independent_property_usage	X			X	X	X				
index_list			X							
index_list_body			X							
index_list_entry			X							
index_list_header			X							
item	X	X	X	X	X	X	X	X		X
item_identification	X	X	X	X	X	X	X	X		X
item_list	X									
item_list_drawing	X									
item_parent_to_item_child_-relationship				X	X					
item_parent_to_tdp_element_-child_relationship				X	X					
item_type	X	X		X	X	X	X			
item_usage	X									
label	X	X	X	X	X	X	X	X	X	X
list_item_usage	X									
list_presentation									X	
load_point									X	

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
locally_defined_item	X									
location_instance	X									
lot_effectivity	X			X	X					
made_from_stock_material	X									
make_from_relationship	X									
measure	X			X	X				X	
notation	X	X	X	X	X					X
number_with_units	X			X	X					
other_list		X								X
other_list_body										X
other_list_body_table										X
other_list_header		X								X
other_reference_document	X	X	X	X	X	X	X	X		X
page_parameters									X	
page_presentation									X	
part_occurrence_in_assembly	X									
parts_list	X	X								
parts_list_body	X									
parts_list_header	X	X								
person	X	X	X	X	X	X	X	X		X
person_and_organization	X	X	X	X	X	X	X			X
product_configuration	X	X	X	X	X	X	X	X		X
product_data_set	X	X		X		X				X
product_data_set_with_format	X	X		X		X				X
product_data_set_with_shading	X	X		X		X				X

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
product_data_set_without_format	X	X		X		X				X
product_model	X	X	X	X	X	X	X			X
product_presentation						X	X			
project	X			X	X					
promissory_usage	X									
property_assignment	X			X	X	X				
quantified_part_usage_in_-assembly	X									
quantity	X			X	X					
reason					X					
reference_document	X	X	X	X	X	X	X	X		X
reference_document_usage	X	X	X	X	X	X	X	X		X
relative_event	X			X	X					
release_authentication	X	X	X	X	X	X	X			X
retrofit_usage	X			X						
revision	X	X	X	X	X					X
security_classification	X	X	X	X	X	X	X			X
sequence_effectivity	X			X	X					
shaded_shape_model						X				
shape_model						X	X			
sheet	X	X		X	X	X	X			X
simple_list_of_elements					X					
simple_list_of_files					X					
single_document_list		X	X							
size_characteristics	X	X	X	X	X	X	X			X

Table 2 — Applicability of types and entities to the units of functionality (continued)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
size_characteristics_full_size	X	X	X	X	X	X	X			X
size_characteristics_internal_-divisions					X	X				
size_characteristics_sheet_-across_file					X	X				
source_file						X				
source_information_type	X			X	X					
source_location		X		X	X					
special_condition	X	X		X	X	X				
specification_document	X	X		X	X			X		
specified_part_in_assembly_tree	X									
standard_document	X	X		X	X			X		
status	X	X	X	X	X	X	X	X		X
stock_material	X									
stock_size	X									
stock_size_parameters	X									
string_location_definition									X	
system_declaration					X	X				
system_destination					X					
tabulation									X	
tabulation_entry										X
tabulation_header									X	
tdp_element	X	X	X	X	X	X	X	X		X
tdp_element_and_item_association	X	X	X	X	X	X	X	X		X
tdp_element_list_item	X									

Table 2 — Applicability of types and entities to the units of functionality (concluded)

Objects	Units of functionality									
Entity	**PL**	**DL**	**IL**	**IDL**	**DDE**	**PDS**	**DWG**	**RD**	**PRES**	**OL**
tdp_element_parent_to_item_-child_relationship				X	X					
tdp_element_parent_to_tdp_-element_child_relationship				X	X					
tdp_indentured_item				X	X					
tdp_indentured_tdp_element				X	X					
text	X	X	X	X	X	X	X	X	X	X
time	X	X	X	X	X	X	X	X	X	X
time_interval_effectivity	X			X	X					
transformation	X									
volume	X			X	X					
weight	X			X	X					

NOTE 4 Table 1 and Table 2 are a cross reference of the UoFs of this part of ISO 10303 and the application objects defined in 4.2.

4.1.1 data_definition_exchange

The data_definition_exchange UoF captures the shipping and packaging information required to identify and relate the product data and documentation being exchanged, in whole or in part, between two enterprises or between two systems.

The following application objects are used by the data_definition_exchange UoF:

— A_number;

— A_real;

— Accessed_file;

— Alternate_element_identification;

— Alternate_identification_of_item;

— Alternate_item;

— An_integer;

— Approval;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Content_property;

— Contract;

— Contract_submission;

— Data_definition_entry_item;

— Data_definition_entry_tdp_element;

— Data_definition_exchange;

— Data_definition_exchange_body;

— Data_definition_exchange_header;

— Data_definition_exchange_simple_entry;

— Data_definition_indentured_entry;

— Data_definition_indentured_list_method;

— Date;

— Date_effectivity;

— Delivery_accounting;

— Design_authority;

— Distribution_notice;

— Document_usage_parameter;

— Drawing_suffix_number_combination;

— Effectivity;

— Element_identification;

— Element_type;

— Event;

— Exchange_file;

— External_library_reference;

— File;

— File_format;

— File_relationship;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Indentured_list_by_document;

— Indentured_list_by_part;

— Indentured_list_by_part_with_document_references_to_parts;

— Independent_property;

— Independent_property_relationship;

— Independent_property_usage;

— Item;

— Item_identification;

— Item_parent_to_item_child_relationship;

— Item_parent_to_tdp_element_child_relationship;

— Item_type;

— Label;

— Lot_effectivity;

— Measure;

— Notation;

— Number_with_units;

— Other_reference_document;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_model;

— Project;

— Property_assignment;

— Quantity;

— Reason;

— Reference_document;

— Reference_document_usage;

— Relative_event;

— Release_authentication;

— Revision;

— Security_classification;

— Sequence_effectivity;

— Sheet;

— Simple_list_of_elements;

— Simple_list_of_files;

— Size_characteristics;

— Size_characteristics_full_size;

— Size_characteristics_internal_divisions;

— Size_characteristics_sheet_across_file;

— Source_information_type;

— Source_location;

— Special_condition;

— Specification_document;

— Standard_document;

— Status;

— System_declaration;

— System_destination;

— Tdp_element;

— Tdp_element_and_item_association;

— Tdp_element_parent_to_item_child_relationship;

— Tdp_element_parent_to_tdp_element_child_relationship;

— Tdp_indentured_item;

— Tdp_indentured_tdp_element;

— Text;

— Time;

— Time_interval_effectivity;

— Volume;

— Weight.

4.1.2 data_list

The data_list UoF contains the information that is formulated into a tabulation of drawings, reference documents, subordinate data list, associated parts list, and other lists pertaining to some Item.

The following application objects are used by the data_list UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— Alternate_identification_of_item;

— An_integer;

— Approval;

— Associated list;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Contract;

— Contract_submission;

— Data_list;

— Data_list_body;

— Data_list_entry;

— Data_list_header;

— Data_list_tabulation;

— Date;

— Design_authority;

— Distribution_notice;

— Document_list;

— Document_usage_parameter;

— Drawing;

— Drawing_suffix_number_combination;

— Element_identification;

— Element_type;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Item;

— Item_identification;

— Item_type;

— Label;

— Notation;

— Other_list;

— Other_list_header;

— Other_reference_document;

— Parts_list;

— Parts_list_header;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_data_set;

— Product_data_set_with_format;

— Product_data_set_with_shading;

— Product_data_set_without_format;

— Product_model;

— Reference_document;

— Reference_document_usage;

— Release_authentication;

— Revision;

— Security_classification;

— Sheet;

— Single_document_list;

— Size_characteristics;

— Size_characteristics_full_size;

— Source_location;

— Special_condition;

— Specification_document;

— Standard_document;

— Status;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.1.3 drawing

The drawing UoF contains the information that describes the presentation of product data in a human interpretable form wherein the physical or functional requirements for that product are presented pictorially, textually, or both.

The following application objects are used by the drawing UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— Alternate_identification_of_item;

— An_integer;

— Annotation;

— Approval;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Contract;

— Date;

— Design_authority;

— Distribution_notice;

— Document_usage_parameter;

— Drawing;

— Drawing_suffix_number_combination;

— Element_identification;

— Element_type;

— Geometry;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Item;

— Item_identification;

— Item_type;

— Label;

— Other_reference_document;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_model;

— Product_presentation;

— Reference_document;

— Reference_document_usage;

— Release_authentication;

— Security_classification;

— Shape_model;

— Sheet;

— Size_characteristics;

— Size_characteristics_full_size;

— Status;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.1.4 indentured_data_list

The indentured_data_list UoF contains the information that provides a configuration-controlled, hierarchical structure of product information. Information may be made up of parts, documents, and parts plus documents.

NOTE A top-down-break-down order is a term sometimes used for a hierarchical structure of product information.

The following application objects are used by the indentured_data_list UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— Alternate_identification_of_item;

— Alternate_item;

— An_integer;

— Approval;

— Associated_list;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Content_property;

— Contract;

— Contract_submission;

— Date;

— Date_effectivity;

— Design_authority;

— Distribution_notice;

— Document_list;

— Document_usage_parameter;

— Drawing;

— Drawing_suffix_number_combination;

— Effectivity;

— Element_identfication;

— Element_type;

— Event;

— External_library_reference;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Indentured_data_list;

— Indentured_data_list_body;

— Indentured_data_list_entry;

— Indentured_data_list_header;

— Indentured_data_list_tabulation;

— Indentured_list_by_document;

— Indentured_list_by_part;

— Indentured_list_by_part_with_document_references_to_parts;

— Independent_property;

— Independent_property_relationship;

— Independent_property_usage;

— Item;

— Item_identification;

— Item_parent_to_item_child_relationship;

— Item_parent_to_tdp_element_child_relationship;

— Item_type;

— Label;

— Lot_effectivity;

— Measure;

— Notation;

— Number_with_units;

— Other_reference_document;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_data_set;

— Product_data_set_with_format;

— Product_data_set_with_shading;

— Product_data_set_without_format;

— Product_model;

— Project;

— Property_assignment;

— Quantity;

— Reference_document;

— Reference_document_usage;

— Relative_event;

— Release_authentication;

— Retrofit_usage;

— Revision;

— Security_classification;

— Sequence_effectivity;

— Sheet;

— Size_characteristics;

— Size_characteristics_full_size;

— Source_information_type;

— Source_location;

— Special_condition;

— Specification_document;

— Standard_document;

— Status;

— Tdp_element;

— Tdp_element_and_item_association;

— Tdp_element_parent_to_item_child_relationship;

— Tdp_element_parent_to_tdp_element_child_relationship;

— Tdp_indentured_item;

— Tdp_indentured_tdp_element;

— Text;

— Time;

— Time_interval_effectivity;

— Volume;

— Weight.

4.1.5 index_list

The index_list UoF contains the information that provides a simple tabulation of data lists and subordinate index lists pertaining to some Item.

The following application objects are used by the index_list UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— An_integer;

— Approval;

— Associated_list;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Contract;

— Data_list;

— Data_list_header;

— Date;

— Design_authority;

— Distribution_notice;

— Document_list;

— Document_usage_parameter;

— Drawing;

— Element_identification;

— Element_type;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Indentured_data_list;

— Indentured_data_list_header;

— Index_list;

— Index_list_body;

— Index_list_entry;

— Index_list_header;

— Item;

— Item_identification;

— Label;

— Notation;

— Other_reference_document;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_model;

— Reference_document;

— Reference_document_usage;

— Release_authentication;

— Revision;

— Security_classification;

— Single_document_list;

— Size_characteristics;

— Size_characteristics_full_size;

— Status;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.1.6 other_list

The other_list UoF contains the information that provides a general structure for simple lists of product information.

The following application objects are used by the other_list UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— An_integer;

— Approval;

— Associated_list;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Contract;

— Date;

— Design_authority;

— Distribution_notice;

— Document_usage_parameter;

— Drawing;

— Element_identification;

— Element_type;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Item;

— Item_identification;

— Label;

— Notation;

— Other_list;

— Other_list_body;

— Other_list_body_table;

— Other_list_header;

— Other_reference_document;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_data_set;

— Product_data_set_with_format;

— Product_data_set_with_shading;

— Product_data_set_without_format;

— Product_model;

— Reference_document;

— Reference_document_usage;

— Release_authentication;

— Revision;

— Security_classification;

— Sheet;

— Size_characteristics;

— Size_characteristics_full_size;

— Status;

— Tabulation_entry;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.1.7 parts_list

The parts_list UoF contains the information that describes a configuration controlled tabulation of all parts and bulk materials consumed within an item. Referenced documents that support the items being describe may also be included in the tabulation.

The following application objects are used by the parts_list UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— Alternate_identification_of_item;

— Alternate_item;

— An_integer;

— Approval;

— Assembly_relationship;

— Associated_list;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Contract;

— Date;

— Date_effectivity;

— Design_authority;

— Distribution_notice;

— Document_usage_parameter;

— Drawing;

— Drawing_suffix_number_combination;

— Effectivity;

— Element_identification;

— Element_type;

— Event;

— External_library_reference;

— File;

— File_format;

— File_relationship

— Foreign_defined_item;

— General_item_definition_relationship;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Independent_property;

— Independent_property_relationship;

— Independent_property_usage;

— Item;

— Item_identification;

— Item_list;

— Item_list_drawing;

— Item_type;

— Item_usage;

— Label;

— List_item_usage;

— Locally_defined_item;

— Location_instance;

— Lot_effectivity;

— Made_from_stock_material;

— Make_from_relationship;

— Measure;

— Notation;

— Number_with_units;

— Other_reference_document;

— Part_occurrence_in_assembly;

— Parts_list;

— Parts_list_body;

— Parts_list_header;

— Person;

— Person_and_organization;

— Product_configuration;

— Product_data_set;

— Product_data_set_with_format;

— Product_data_set_with_shading;

— Product_data_set_without_format;

— Product_model;

— Project;

— Promissory_usage;

— Property_assignment;

— Quantified_part_usage_in_assembly;

— Quantity;

— Reference_document;

— Reference_document_usage;

— Relative_event;

— Release_authentication;

— Retrofit_usage;

— Revision;

— Security_classification;

— Sequence_effectivity;

— Sheet;

— Size_characteristics;

— Size_characteristics_full_size;

— Source_information_type;

— Special_condition;

— Specification_document;

— Specified_part_in_assembly_tree;

— Standard_document;

— Status;

— Stock_material;

— Stock_size;

— Stock_size_parameters;

— Tdp_element;

— Tdp_element_and_item_association;

— Tdp_element_list_item;

— Text;

— Time;

— Time_interval_effectivity;

— Transformation;

— Volume;

— Weight.

4.1.8 presentation

The presentation UoF specifies the presentation of geometry and text with attributes. The association of these attributes to the geometry or text is defined either directly or through a context.

EXAMPLE Colour, line thickness or weight, line font are attribute examples.

NOTE 1 Visual presentation can be for data content defined within this part of ISO 10303 or for additional data that is not explicitly specified in this part of ISO 10303.

NOTE 2 ISO 8879 Standard Generalized Markup Language (SGML)[1], ISO 10744 HyTime[2], and ISO 10179 Document Style Semantics and Specification Language[3], satisfy presentation of associated lists with relationships into this part of ISO 10303.

The following application objects are used by the presentation UoF:

— A_number;

— A_real;

— An_integer;

— Body_presentation;

— Column_header;

— Columnar_data_content_holder;

— Date;

— Explicit_graphics;

— External_graphics_file;

— Field_content_holder;

— Field_content_holder_singular;

— Field_content_holder_tabulation;

— Field_location;

— Field_location_definition;

— Field_title;

— File;

— File_format;

— File_relationship;

— Geometry;

— Group_parent_column_header;

— Group_parent_content_holder;

— Group_parent_header;

— Header_block;

— Header_presentation;

— Identifier;

— Label;

— List_presentation;

— Load_point;

— Measure;

— Page_parameters;

— Page_presentation;

— String_location_definition;

— Tabulation;

— Tabulation_header;

— Text;

— Time.

4.1.9 product_data_set

The product_data_set UoF contains the information that describes a group of related product information about an item.

EXAMPLE Shape model, NC programs and FEA models are examples.

The following application objects are used by the product_data_set UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— Alternate_identification_of_item;

— An_integer;

— Annotation;

— Approval;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Configuration;

— Content_property;

— Contract;

— Date;

— Design_authority;

— Distribution_notice;

— Document_usage_parameter;

— Drawing;

— Drawing_suffix_number_combination;

— Element_identification;

— Element_type;

— External_library_reference;

— File;

— File_format;

— File_relationship;

— Geometric_validation_property;

— Geometry;

— Header;

— Header_configuration_with_element_identification;

— Identifier;

— Independent_property;

— Independent_property_relationship;

— Independent_property_usage;

— Item;

— Item_identification;

— Item_type;

— Label;

— Other_reference_document:

— Person;

— Person_and_organization;

— Product_configuration;

— Product_data_set;

— Product_data_set_with_format;

— Product_data_set_with_shading;

— Product_data_set_without_format;

— Product_model;

— Product_presentation;

— Property_assignment;

— Reference_document;

— Reference_document_usage;

— Release_authentication;

— Security_classification;

— Shaded_shape_model;

— Shape_model;

— Sheet;

— Size_characteristics;

— Size_characteristics_full_size;

— Size_characteristics_internal_divisions;

— Size_characteristics_sheet_across_file;

— Source_file;

— Special_condition;

— Status;

— System_declaration;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.1.10 reference_document

The Reference_document UoF identifies documents and their usages that are required as portions of the information set related to the product.

The following application objects are used by the reference_document UoF:

— A_number;

— A_real;

— Alternate_element_identification;

— An_integer;

— Certification;

— Change_identification;

— Company;

— Company_code;

— Date;

— Design_authority;

— Document_usage_parameter;

— Element_identification;

— Element_type;

— Identifier;

— Item;

— Item_identification;

— Label;

— Other_reference_document;

— Person;

— Product_configuration;

— Reference_document;

— Reference_document_usage;

— Specification_document;

— Standard_document;

— Status;

— Tdp_element;

— Tdp_element_and_item_association;

— Text;

— Time.

4.2 Application objects

This subclause specifies the application objects for the technical data packaging core information and exchange application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

NOTE Application Object EXPRESS:

```
*)
SCHEMA AP232_ARM;
(*
```

4.2.1 A_number

A_number is a numeric value that quantifies something.

EXAMPLE 1 110 is a number that specifies the voltage of an electric motor.

EXAMPLE 2 2.5 is a number that represents the size diameter of a bar of stock.

NOTE Application Object EXPRESS:

```
*)
TYPE a_number = NUMBER;
END_TYPE;
(*
```

4.2.2 A_real

An A_real is a numeric value that quantifies something. The A_real domain is all rational, irrational and scientific real numbers.

NOTE Application Object EXPRESS:

```
*)
TYPE a_real = REAL;
END_TYPE;
(*
```

4.2.3 Accessed_file

An Accessed_file is a type of File (see 4.2.59) that specifies the information needed to locate a digital file in a remote location, where at least one of the data elements for Accessed_file shall exist.

NOTE 1 This form of exchanging or delivering data is referred to as "delivery of access" where TDP data is not physically exchanged, but rather information on where to locate the data is exchanged.

The data associated with an Accessed_file are the following:

— path_information;

— storage_node_identification.

NOTE 2 Application Object EXPRESS:

```
*)
ENTITY accessed_file
SUBTYPE OF (file);
  storage_node_identification : OPTIONAL text;
  path_information : OPTIONAL text;
WHERE
  WR1: EXISTS (storage_node_identification) or
  EXISTS (path_information);
END_ENTITY;
(*
```

4.2.3.1 path_information

The path_information specifies the appropriate drive, directory, node or other location information required for the operating system software application environment within the network node or device. The path_-information need not be specified for a particular Accessed_file.

4.2.3.2 storage_node_identification

The storage_node_identification specifies the network protocol address of the network node or device. The storage_node_identification need not be specified for a particular Accessed_file.

4.2.4 Alternate_element_identification

The Alternate_element_identification is an additional identification for a specific group of product information. An Alternate_element_identification may be used to identify a specific group of product information instead of using an Element_identification (see 4.2.46). Additional configuration control identification information may be associated with the Alternate_element_identification.

NOTE 1 A specific group of product information may be a TDP element, or an other product information grouping defined outside this standard. A specific group of product information may be configuration controlled as a paper document, a single file, multiple files, or within a data base.

NOTE 2 Alternate_element_identification allows a Tdp_element (for example, document) to be initially created and managed in some other organization and then managed in another organization using a different set of identification information.

EXAMPLE A document from one company (supplier) is used in other company but the second company (assembly manufacture) may change its identification so information would work with in a particular software application. This software application or system might have its own naming convention and classification structure. This second identification would need to be managed and controlled.

The data associated with an Alternate_element_identification are the following:

— change_status;

— design_activity;

— element_classifications;

— identifying_number;

— outstanding_changes;

— title.

NOTE 3 Application object EXPRESS:

```
*)
ENTITY alternate_element_identification;
  change_status : OPTIONAL change_identification;
  design_activity : design_authority;
  element_classifications: OPTIONAL LIST [1:?] OF element_type;
  identifying_number : identifier;
  outstanding_changes : OPTIONAL LIST [1:?] OF other_reference_document;
  title : OPTIONAL text;
END_ENTITY;
(*
```

4.2.4.1 change_status

The change_status specifies the identification of the most recent change information that is recorded for the Alternate_element_identification of interest. The change_status need not be specified for a particular Alternate_element_identification.

4.2.4.2 design_activity

The design_activity specifies the organization, company, business, or industry responsible for the Alternate_element_identification.

4.2.4.3 element_classifications

The element_classifications specifies a classification code for the specific group of product information of interest. The classification code is utilized for data management of the specific group of product information of interest. The element_classifications need not be specified for a particular Alternate_-element_identification. There may be more than one element_classifications for an Alternate_element_-identification.

EXAMPLE "PL" is an element_classification for a Parts list.

4.2.4.4 identifying_number

The identifying_number specifies an identification number for the specific group of product information of interest as issued by the design_activity.

NOTE The identifying_number may be an alphanumeric.

4.2.4.5 outstanding_changes

The outstanding_changes specifies the identification of one or more changes to a specific group of product information of interest that have not been incorporated. The outstanding_changes need not be specified for a particular Alternate_element_identification. There may be more than one outstanding_changes for an Alternate_element_identification.

4.2.4.6 title

The title specifies the formal designation of the specific group of product information of interest. The title need not be specified for a particular Alternate_element_identification.

NOTE The document title may be an abbreviated form of the title or may have abbreviations of selected words within the title.

4.2.5 Alternate_identification_of_item

An Alternate_identification_of_item is an additional identification of an Item (see 4.2.89) for purposes of configuration control. An Alternate_identification_of_item may be used to identify an Item instead of an Item_identification (see 4.2.90).

NOTE 1 Alternate_identification_of_item allows an Item (part) to be initially created and managed in some other organization and then managed in another organization using a different set of identification information.

EXAMPLE A part from one company (supplier) is used in other company but the second company (assembly manufacture) may change its identification so information would work with in a particular software application. This software application or system might have its own naming convention and classification structure. This second identification would need to be managed and controlled.

The data associated with an Alternate_identification_of_item are the following:

— change_status;

— classifications;

— design_activity;

— identifying_number;

— nomenclature_or_name;

— source_information.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY alternate_identification_of_item;
  change_status : OPTIONAL change_identification;
  classifications : OPTIONAL LIST [1:?] of label;
  design_activity : design_authority;
  identifying_number : item_identifying_number_select;
  nomenclature_or_name : OPTIONAL label;
  source_information : OPTIONAL source_information_type;
END_ENTITY;
(*
```

4.2.5.1 change_status

The change_status specifies the current level of change activity that an Item_identification has undergone. The change_status need not be specified for a particular Alternate_identification_of_item.

NOTE If the change_status is not specified, then the Item is at the original release or issue.

4.2.5.2 classifications

The classifications specifies a set of product types or categories for an Item. The classifications need not be specified for a particular Alternate_identification_of_item. There may be more than one classifications for an Alternate_identification_of_item.

4.2.5.3 design_activity

The design_activity specifies the design authority responsible for the Item..

4.2.5.4 identifying_number

The identifying_number specifies an Identifier or a Drawing_suffix_number_combination that is an identifier that defines the identification number for the Item as issued by the design_activity of the component.

NOTE 1 Industrial practices employ a practice of concatenating a dash and an alphanumeric to a drawing number to identify an Item. Each Item receiving a unique alphanumeric for that drawing.

NOTE 2 Application object EXPRESS:

```
*)
TYPE item_identifying_number_select = SELECT
  (identifier,
  drawing_suffix_number_combination);
END_TYPE;
(*
```

4.2.5.5 nomenclature_or_name

The nomenclature_or_name specifies the name, noun phrase, or abbreviated name of the Item. The nomenclature_or_name need not be specified for a particular Alternate_identification_of_item.

4.2.5.6 source_information

The source_information specifies the method for item procurement. The source_information need not be specified for a particular Alternate_identification_of_item.

4.2.6 Alternate_item

An Alternate_item is an Item (see 4.2.89) that can be used in lieu of the original Item. An Alternate_item is interchangeable with another Item with respect to form, fit, and function defined by the original Item within the context of the usage conditions defined by the Alternate_item. The data associated with an Alternate_item are the following:

— interchange_item;

— preference_order;

— type_of_alternate;

— usage_conditions.

NOTE Application Object EXPRESS:

```
*)
ENTITY alternate_item;
  interchange_item: usage_context;
  preference_order : OPTIONAL an_integer;
  type_of_alternate: OPTIONAL label;
  usage_conditions: OPTIONAL LIST [1:?]OF text;
END_ENTITY;
(*
```

4.2.6.1 interchange_item

The interchange_item specifies the Item that can be used in lieu of the original_item. The substitutability of the interchange_item may be constrained by its usage. The usage context of the interchange_item provides the ability to apply usage parameter through an Item_usage.

NOTE Application object EXPRESS:

```
*)
TYPE usage_context = SELECT
  (item,
  item_usage);
END_TYPE;
(*
```

4.2.6.2 preference_order

The preference_order specifies An_integer that defines the order of preference in the usage of the Alternate_item relative to another Alternate_item of the original Item. The preference_order need not be specified for a particular Alternate_item.

4.2.6.3 type_of_alternate

The type_of_alternate specifies the term used for the category, class, or type of alternate. The type_of_-alternate need not be specified for a particular Alternate_item.

NOTE A type_of_alternate may be an acceptable replacement or substitute of the original Item.

4.2.6.4 usage_conditions

The usage_conditions specifies the conditions or parameters under which the Alternate_item is used. The usage_conditions are defined in terms of a specific usage of the Alternate_item in the context of the original Item. The usage_conditions need not be specified for a particular Alternate_item. There may be more than one usage_condition for an Alternate_item.

4.2.7 An_integer

An_integer is a numeric value that quantifies something. The An_integer domain is all positive integer numbers.

NOTE Application object EXPRESS:

```
*)
TYPE an_integer = INTEGER;
END_TYPE;
(*
```

4.2.8 Annotation

The Annotation is the text, geometry, or symbology that is applied to product information for human interpretation. The data associated with an Annotation are the following:

— annotation_geometry.

NOTE Application object EXPRESS:

```
*)
ENTITY annotation;
  annotation_geometry : OPTIONAL geometry;
END_ENTITY;
(*
```

4.2.8.1 annotation_geometry

Annotation_geometry specifies placement, orientation, location, and size of geometry for the Annotation. The annotation_geometry need not be specified for a particular Annotation.

EXAMPLE 1 Borders for columns and rows of tables.

EXAMPLE 2 Surface finish symbols.

4.2.9 Approval

An Approval is the indication, within an organization, of concurrence with a piece of data. The data associated with an Approval are the following:

— approval_date;

— person_organization;

— purpose_of;

— status.

NOTE Application object EXPRESS:

```
*)
ENTITY approval;
  approval_date : date;
  purpose_of : text;
  person_organization : person_and_organization;
  status : text;
END_ENTITY;
(*
```

4.2.9.1 approval_date

The approval_date specifies when the Approval is effective.

4.2.9.2 person_organization

The person_organization specifies the person and organization that has granted the Approval.

4.2.9.3 purpose_of

The purpose_of specifies the reason for the Approval.

4.2.9.4 status

The status specifies the state of consent applied to product data.

4.2.10 Assembly_relationship

An Assembly_relationship is a type of Locally_defined_item (see 4.2.101) specifying that the component Item (see 4.2.89) which the Item_usage (see 4.2.96) is being defined for is a portion of an assembly. The different collection of Item(s) in an assembly can be connected, fastened, joined, coupled, bound, or associated for purposes of identification. An assembly is one of the states an Item can be while it plays the role of a component during a particular Item_usage.

NOTE 1 The collection of Items that make up an assembly is captured through the Item_list entity. The Assembly_relationship object defined in this part of ISO 10303 specifies the relationship between an assembly and one of the assembly components. When an Item_usage is an Assembly_relationship, then the corresponding Item_-list is considered as a particular definition of an assembly. An Assembly_relationship usually conveys a particular usage, such as a next assembly usage, a specified higher usage, or promissory usage. An Assembly_relationship can also convey an instance of a component of an assembly.

NOTE 2 The collection of components may be a mechanical assembly, an electrical assembly, or a single component that is being re-identified for purposes of configuration control.

NOTE 3 Application object EXPRESS:

```
*)
ENTITY assembly_relationship
  SUPERTYPE OF (ONEOF (quantified_part_usage_in_assembly,
     part_occurrence_in_assembly, promissory_usage,
      specified_part_in_assembly_tree))
  SUBTYPE OF (locally_defined_item);
END_ENTITY;
(*
```

4.2.11 Associated_list

An Associated_list is a type of Tdp_element (see 4.2.170) that is a configuration controlled tabulation of pertinent engineering information pertaining to an Item or a Tdp_element. Each Associated_list is either a Document_list (see 4.2.41), a Parts_list (see 4.2.117), or an Other_list (see 4.2.109). The Item may be depicted on a drawing or in a set of drawings and the presentation information required to graphically present the information to a person may be included.

NOTE 1 Engineering information consist of parts lists, list of materials, list of drawings, list of reference documents, product data sets, and other subordinate data that pertains to the Item to which the TDP element applies.

The data associated with an Associated_list are the following:

— presentation;

— related_to.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY associated_list
  ABSTRACT SUPERTYPE OF (ONEOF(document_list,parts_list,other_list))
  SUBTYPE OF (tdp_element);
  presentation : OPTIONAL list_presentation;
  related_to : OPTIONAL drawing_or_product_data_set;
END_ENTITY;
(*
```

4.2.11.1 presentation

The presentation specifies the graphical information utilized to facilitate human interpretation of the Associated_list. The presentation need not be specified for a particular Associated_list.

4.2.11.2 related_to

The related_to specifies a Drawing or a Product_data_set that the Associated_list is associated. The related_to need not be specified for a particular associated_list.

NOTE Application object EXPRESS:

```
*)
TYPE drawing_or_product_data_set = SELECT
  (drawing,
   product_data_set);
END_TYPE;
(*
```

4.2.12 Body_presentation

The Body_presentation is the viewable form of the main part of the document. It contains the information necessary to format the data in the main part of the document. There may be two sets of body information, one for the first page and one for each remaining page of the document. The data associated with a Body_-presentation are the following:

— body_tabulation_presentation.

NOTE Application object EXPRESS:

```
*)
ENTITY body_presentation;
  body_tabulation_presentation : LIST [1:?] OF tabulation;
END_ENTITY;
(*
```

4.2.12.1 body_tabulation_presentation

The body_tabulation_presentation specifies the table layout information for the data contained in the body of the document. There may be more than one body_tabulation_presentation for a Body_presentation.

NOTE All body data is considered tabular.

4.2.13 Certification

The Certification is the qualification of product data.

NOTE A supplier certification can be obtained through the alternate identification of an Item (see 4.2.89) or a TDP_element.

The data associated with a Certification are the following:

— approvals;

— creation_date;

— description;

— name.

NOTE Application object EXPRESS:

```
*)
ENTITY certification;
  approvals : OPTIONAL SET[1:?] OF approval;
  creation_date : OPTIONAL date;
  description : OPTIONAL text;
  name : label;
END_ENTITY;
(*
```

4.2.13.1 approvals

The approvals specifies the certification is confirmed valid. There may be more than one approvals for a Certification. The approvals need not be specified for a particular Certification.

4.2.13.2 creation_date

The creation_date specifies the date the certification was formulated. The creation_date need not be specified for a particular Certification.

4.2.13.3 description

The description specifies the text that characterizes the Certification. The descrption need not be specified for a particular Certification.

4.2.13.4 name

The name specifies the label by which the Certification is known.

4.2.14 Change_identification

The Change_identification is the specified information for the current level of incorporated changes for a TDP element. The data associated with a Change_identification are the following:

— action_status;

— assigned_to;

— change_code;

— change_date;

— change_description;

— change_level;

— issue_date;

— issue_level;

— revision_authorization_identifications;

— revision_date;

— revision_description;

— revision_level.

NOTE Application object EXPRESS:

```
*)
ENTITY change_identification;
     action_status : OPTIONAL status;
     assigned_to : OPTIONAL SET[1:?] OF person_and_organization;
     change_code : OPTIONAL change_type;
     change_date : OPTIONAL date;
     change_description : OPTIONAL text;
     change_level : OPTIONAL identifier;
     issue_date : OPTIONAL date;
     issue_level : OPTIONAL identifier;
     revision_authorization_identifications : OPTIONAL SET [1:?] OF
     revision_authorization_select;
     revision_date : OPTIONAL date;
     revision_description : OPTIONAL text;
     revision_level : OPTIONAL identifier;
WHERE
     WR1:       EXISTS (revision_level) OR
       EXISTS (revision_date) OR EXISTS (change_date) OR
       EXISTS (change_level) OR
       EXISTS (issue_level) OR EXISTS (issue_date);
END_ENTITY;
(*
```

4.2.14.1 action_status

The action_status specifies the life cycle stage or the working process stage of the TDP element being identified. The action_status need not be specified for a particular Change_identification.

NOTE Action_status is the identification of the process stage the TDP element is in during the life cycle of creating and supporting product information.

EXAMPLE In-work; Complete, In-review, and Final, are examples of action_status.

4.2.14.2 assigned_to

The assigned_to specifies a person and organization that has an association with the change activity. The assigned_to need not be specified for a particular Change_identification. There may be more than one assigned_to for a Change_identification.

4.2.14.3 change_code

The change_code specifies the type of change identified by the revision_level and the incorporated changes at that change_level. The change_code need not be specified for a particular Change_identification.

NOTE 1 Table 3 is the relationship of Change_identification attributes for typical TDP industrial practices.

Table 3 — Change_identification attribute relationships

Change_-identification.change_code	Meaning	Change_-identification.revision_level	Change_-identification.change_level
Original	The File,Item, or Tdp_-element is the original.	Not defined or 0	Not defined or 0
Orginal_with_change	The File, Item, or Tdp_-element is the original with changes incorporated.	Not defined or 0	Highest level change included
Complete_revision	The File, Item, or Tdp_-element is a complete revision to a document or product data.	Indicates current revision level	Not defined or 0
Revision_with_-change	The File,Item, or Tdp_-element is a complete revision of a file, item, or Tdp_-element with changes incorporated.	Indicates current revision level	Highest level change included
Change	The File, Item, or Tdp_-element is change material to an original, revision, or supplement.	Not defined or 0	Highest level change included
Other	Used when none of the above change types apply	Not defined or 0	Not defined or 0

The value of the change_code shall be one of the following:

— change;

— complete_revision;

— original;

— original with change;

— other;

— revision with change.

NOTE 2 See 4.2.14.3 - 4.2.14.3.4 for the definition of each allowable value of change_code.

NOTE 3 Application object EXPRESS:

```
*)
TYPE change_type = ENUMERATION OF
  (change,
  complete_revision,
  original,
  original_with_change,
  other,
  revision_with_change);
END_TYPE;
(*
```

4.2.14.3.1 change
The change_identification is a File, Item, or Tdp_element that modifies an original, revision, or supplement.

NOTE Revision level is typically not defined for an original or is defined as zero. The change level is the highest level change included.

4.2.14.3.2 complete_revision
The File, Item, or Tdp_element that is a complete revision to a File, Item, or Tdp_element.

NOTE Revision_level indicates current revision level. Change_level is typically zero for a complete revision.

4.2.14.3.3 original
The File, Item, or Tdp_element that is the original release.

4.2.14.3.4 original with change
The File, Item, or Tdp_element that is the original with changes incorporated.

NOTE Revision level is typically not defined for an original or is defined as zero. The change level is highest change included.

4.2.14.3.5 other
a modification where none of the other Change_codes apply.

4.2.14.3.6 revision with change
The File, Item, or Tdp_element that is a complete revision of a document or product data set with changes incorporated.

NOTE Revision_level is current revision level. Change_level is highest change included.

4.2.14.4 change_date

The change_date specifies either the date and time that the change_level was advanced or the date and time of the change. The change_date need not be specified for a particular Change_identification.

EXAMPLE Figure 4 illustrates the format style for change_date.

YYYYMMDD/HHHH:SS
YYYY - Year
MM - Month
DD - Day
HHHH - Hour
SS - Second

Figure 4 — Format style for date

4.2.14.5 change_description

The change_description specifies the text that characterizes the change to the File, Item, or TDP_element. The change_description need not be specified for a particular Change_identification.

EXAMPLE A radius filler's dimension value was increased to increase manufacturability.

4.2.14.6 change_level

The change_level specifies the latest alteration made to a File, Item, or Tdp_ element as part of a revision. The change_level need not be specified for a particular Change_identification.

EXAMPLE An incorporated change from Drawing Change Notice identified as change level "C".

4.2.14.7 issue_date

The issue_date specifies the date and time that the activity issued the File, Item, or Tdp_element. The issue_date need not be specified for a particular Change_identification.

EXAMPLE Figure 4 illustrates the format style for issue_date.

4.2.14.8 issue_level

The issue_level specifies the issue number of the File, Item , or Tdp_element. The issue_level need not be specified for a particular Change_identification.

NOTE 1 Changes to documents are normally tracked by revisions and engineering change notices. However, in cases where changes to the document do not affect technical content, a new issue of the document may be provided for which no revision or engineering change notice level is advanced.

NOTE 2 In product data management systems a sequence number identification is some times generated for each check-in and check-out of a managed item in the system. This unique sequence number is an other type of issue_-level.

4.2.14.9 revision_authorization_identifications

The revision_authorization_identifications specifies an Identifier or an Other_reference_document that is the identification of the revision authorization document. The revision_authorization_identifications can also specify an Approval for the change activity or Contract for identifying under what contract the change was made.. The revision_authorization_ identifications need not be specified for a particular Change_-identification.

NOTE 1 In industrial practices, a revision authorization document is used in lieu of a revision description or a revision record.

NOTE 2 The revision authorization document may describe the revision history of the file, item, or Tdp_element since the original release of the File, Item, or Tdp_element.

NOTE 3 The revision authorization document may describe a single revision of the File, Item, or Tdp_element.

NOTE 4 The Other_reference_document specifies that the Revision_authorization_select is a Reference_ document that defines the rationale for the Change_identification.

NOTE 5 The identifier specifies that the Revision_authorization_select is other than a Other_ reference_-document and identifies the authorizing article.

NOTE 6 Application object EXPRESS:

```
*)
TYPE revision_authorization_select = SELECT
  (approval,
   contract,
   identifier,
   other_reference_document);
END_TYPE;
(*
```

4.2.14.10 revision_date

The revision_date specifies either the date and time that the revision_level see was advanced or the date and time of the revision. The revision_date need not be specified for a particular Change_identification.

EXAMPLE Figure 4 illustrates the format style for revision_date.

4.2.14.11 revision_description

The revision_description specifies the text that characterizes the revision to the File, Item, or Tdp_ element. The revision_description need not be specified for a particular Change_identification.

EXAMPLE The part’s complex surface shape changed due to analysis relsults.

4.2.14.12 revision_level

The revision_level specifies the version of the File, Item, or Tdp_element from its original release or issue. Revisions reflect changes made to the original after authorized release or issue that require the revision level to be advanced. The revision_level need not be specified for a particular Change_identification.

4.2.15 Column_header

The Column_header is the information to define the table's column header text, style, and placement location. Each table on the document may have one or more column headers. The data associated with a Column_header are the following:

— single_column_header_titles.

NOTE Application object EXPRESS:

```
*)
ENTITY column_header;
  single_column_header_titles : LIST [1:?] OF field_title;
END_ENTITY;
(*
```

4.2.15.1 single_column_header_titles

The single_column_header_titles specifies the title text for the Column_header, that specifies the formatting and placement information. There may be more than one single_column_header_titles for a particular column_header.

4.2.16 Columnar_data_content_holder

The Columnar_data_content_holder is the information necessary to place the data contained in the column on the document and identification of the data contents for that column.

NOTE 1 The data to be instantiated in the columns will be defined elsewhere in this part of ISO 10303.

The data associated with a Columnar_data_content_holder are the following:

— column_content_presentation.

NOTE Application object EXPRESS:

```
*)
ENTITY columnar_data_content_holder;
  column_content_presentation : field_content_holder_tabulation;
END_ENTITY;
(*
```


4.2.16.1 column_content_presentation

The column_content_presentation specifies the data content from the list information and the formatting and placement information necessary for instantiation.

4.2.17 Company

A Company is an association of persons that together have a relationship for the operation of a commercial or industrial enterprise. A company shall be identifiable by a name or name and location. The data associated with a Company are:

— address;

— name.

NOTE Application object EXPRESS:

```
*)
ENTITY company;
  address: OPTIONAL text;
  name: LABEL;
END_ENTITY;
(*
```

4.2.17.1 address

The address specifies the electronic or physical location where the company can be interfaced with. The address need not be specified for a particular Company.

EXAMPLE An example of an address may be a mailing address such as the following:

ABC Corporation
523 East Street
Fall City, Missouri 63102
USA

4.2.17.2 name

The name specifies the textual label the company is known by.

4.2.18 Company_code

The Company_code is the identification of the company in computer interpretable form. The Company_-code is a method of identification. The data associated with a Company_code are the following:

— code;

— code_administrator;

— type_of_code.

NOTE Application object EXPRESS:

```
*)
ENTITY company_code;
  code : identifier;
  code_administrator : OPTIONAL company;
  type_of_code : text;
END_ENTITY;
(*
```

4.2.18.1 code

The code specifies the Company referenced within the context of a company identification coding scheme.

NOTE The code is typically unique within the context of a company identification coding scheme.

4.2.18.2 code_administrator

A code_administrator specifies the organization that defines and issues codes for different companies within an industry. A code_administrator need not be specified for a particular Company_code.

4.2.18.3 type_of_code

The type_of_code specifies the name of the company identification coding scheme. The value of the type_-of_code shall be either a human_interpretable string of characters or a computer generated code administered by a code_administrator.

4.2.19 Configuration

A Configuration is the information pertinent to the control and management of an Item (see 4.2.89), a Tdp_element (see), File (see 4.2.59), or a data exchange. The data associated with a Configuration are the following:

— data_usage_rights;

— distribution_authorizations;

— end_item_system_designation;

— preparing_contracts;

— release_authorizations;

— security_identifications.

NOTE Application object EXPRESS:

```
*)
ENTITY configuration;
     data_usage_rights : OPTIONAL label;
     distribution_authorizations : OPTIONAL LIST [1:?] OF
       distribution_notice;
     end_item_system_designation : OPTIONAL product_configuration;
     preparing_contracts : OPTIONAL LIST [1:?] OF contract;
     release_authorizations : OPTIONAL LIST [1:?] OF
       release_authentication;
     security_identifications : OPTIONAL LIST [1:?] OF
       security_classification;
WHERE
     WR1: EXISTS (preparing_contracts) OR
        EXISTS (end_item_system_designation) OR
        EXISTS (release_authorizations) OR
        EXISTS (security_identifications) OR
        EXISTS (distribution_authorizations) OR
        EXISTS (data_usage_rights);
END_ENTITY;
(*
```

4.2.19.1 data_usage_rights

The data_usage_rights specifies if the user of the data has legal rights to use the configuration data as the user sees fit. The data_usage_rights need not be specified for a particular Configuration.

NOTE 1 Limited rights indicates the user has to obtain legal rights from the originating data source to use the data. Unlimited rights indicate the user does not have to obtain legal rights to use the data.

NOTE 2 In industrial practices, data rights (and data rights codes) are specified within the context of the contract that the data was prepared for.

4.2.19.2 distribution_authorizations

The distribution_authorizations specifies the allowable circulation of information that is applicable to the Configuration. The distribution_authorizations need not be specified for a particular Configuration. There may be more than one distribution_authorizations for a Configuration.

4.2.19.3 end_item_system_designation

The end_item_system_designation specifies the product that is applicable to the Configuration. The end_-item_system_designation need not be specified for a particular Configuration.

4.2.19.4 preparing_contracts

The preparing_contracts specifies the contracts authorizing the developing of the Configuration of interest. The preparing_contracts need not be specified for a particular Configuration. There may be more than one preparing_contracts for a configuration.

4.2.19.5 release_authorizations

The release_authorizations specifies the originating system sources that have authenticated the Configuration contents for the purposes of release. The release_authorizations need not be specified for a particular Configuration. There may be more than one release_authorizations for a Configuration.

4.2.19.6 security_identifications

The security_identifications specifies the security or sensitivity of the Configuration and its contents. The security_identification need not be specified for a particular Configuration. There may be more than one security_identifications for a Configuration.

EXAMPLE 1 Security_identifications are TOP SECRET, SECRET, CLASSIFIED, UNCLASSIFIED.

EXAMPLE 2 Company security_identifications are PRIVATE, COMPETITION SENSITIVE, or PROPRIETARY.

4.2.20 Content_property

A Content_property is characteristics about the the kind of information that is being captured in the Tdp_ element (see 4.2.170). The data associated with a Content_property are the following:

— detail_level;

— geometry_type;

— languages;

— real_world_scale.

NOTE Application object EXPRESS:

```
*)
ENTITY content_property;
  detail_level : OPTIONAL text;
  geometry_type : OPTIONAL text;
  languages : OPTIONAL SET [1:?] OF text;
  real_world_scale : OPTIONAL a_real;
WHERE
  WR: EXISTS(detail_level) OR
      EXISTS(geometry_type) OR
      EXISTS(languages) OR
      EXISTS(real_world_scale);
END_ENTITY;
(*
```

4.2.20.1 detail_level

The detail_level specifies the level of information detail that a document or document representation contains. The detail_level need not be specified for a particular Content_property.

4.2.20.2 geometry_type

The geometry_type specifies the kind of geometry that a document contains. The geometry_type need not be specified for a particular Content_property.

4.2.20.3 languages

The languages specifies the languages used in the document. The languages need not be specified for a particular Content_property. There may be more than one languages for a Content_property.

4.2.20.4 real_world_scale

The real_world_scale specifies the scale that is used in the document definition the Content_property is referred by. The real_world_scale need not be specifed for a particular Content_property.

4.2.21 Contract

A Contract is information pertinent to a formal agreement between two or more parties. The data associated with a Contract are the following:

— affected_organizations;

— approvals;

— contract_data_requirements_list;

— contract_number;

— creation_date;

— data_item_description_identification.

NOTE Application object EXPRESS:

```
*)
ENTITY contract;
  affected_organizations : OPTIONAL SET [1:?] OF
          person_and_organization;
  approvals : OPTIONAL LIST [1:?] OF approval;
  contract_data_requirements_list : OPTIONAL identifier;
  contract_number : identifier;
  creation_date : OPTIONAL date;
  data_item_description_identification : OPTIONAL identifier;
END_ENTITY;
(*
```

4.2.21.1 affected_organizations

The affected_organizations specifies the organizations that the contract applies to or the organizations that must work under the conditions set forth in the contract. The affected_organizations need not be specified for a particular Contract.

4.2.21.2 approvals

The approvals specifies the act of confirming officially the contract for some use or for some level of its completeness. The approvals need not be specified for a particular Contract.

4.2.21.3 contract_data_requirements_list

The contract_data_requirements_list specifies the contract data requirement list clause or contract work listing number that authorizes, specifies, and regulates the formal agreement between two or more parties. The contract data requirements list is part of the Contract. The contract_data_requirements_list need not be specified for a particular Contract.

4.2.21.4 contract_number

The contract_number specifies the contract number used to reference the Contract.

NOTE The contract number may be an alphanumeric identification.

4.2.21.5 creation_date

The creation_date specifies the date and time the contract was created. The data_item_description_-identification need not be specified for a particular Contract.

4.2.21.6 data_item_description_identification

The data_item_description_identification specifies a specific set of data or system products that are referenced within a contract. The data_item_description_identification need not be specified for a particular Contract.

4.2.22 Contract_submission

Contract_submission is the identification of when and where a File (see 4.2.59), Item (see 4.2.89), or Tdp_ element (see 4.2.170) was submitted on a Contract (see 4.2.21). The data associated with a Contract_-submission are the following:

— date_of_submission;

— delivered_contract;

— location.

NOTE Application object EXPRESS:

```
*)
ENTITY contract_submission;
  date_of_submission : date;
  delivered_contract : contract;
  location : text;
END_ENTITY;
(*
```

4.2.22.1 date_of_submission

The date_of_submission specifies the date that the File, Item, or Tdp_element was submitted to the location.

4.2.22.2 delivered_contract

The delivered_contract specifies the Contract that the File, Item, or Tdp_element was submitted under.

NOTE The delivery contract for a File, Item, or Tdp_element may or may not be the contract that the File, Item, or Tdp_element was prepared under.

4.2.22.3 location

The location specifies the mailing address, network node, or other identification information required to identify the location where the File, Item, or Tdp_element was submitted to.

4.2.23 Data_definition_entry_item

The Data_definition_entry_item is a collection of characteristics about an item in a data exchange.

NOTE 1 The collection of characteristics are typical characteristics about the data that is maintained within Product Data Management or Configuration Control systems. The characteristics are employed to manage the use and distribution of the File, Item, or Tdp_element.

The data associated with a data_definition_entry_item are the following:

— available_from;

— delivery_accounting_reference;

— entry_files;

— entry_item_change_level;

— entry_notes;

— special_conditions;

— superseded_entry.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY data_definition_entry_item;
  available_from : OPTIONAL LIST [1:?] OF source_identification;
  delivery_accounting_references : OPTIONAL SET [1:?] OF
    delivery_accounting;
  entry_files : OPTIONAL LIST [1:?] OF data_definition_file_entry;
  entry_item_change_level : OPTIONAL LIST [1:?] OF identifier;
  entry_notes : OPTIONAL LIST [1:?] OF notation;
  special_conditions : OPTIONAL LIST [1:?] OF special_condition;
  superseded_entry : OPTIONAL item_identification;
WHERE
  WR1:EXISTS(available_from)
  OR EXISTS(delivery_accounting_references)
  OR EXISTS(entry_files)
  OR EXISTS(entry_item_change_level)
  OR EXISTS(entry_notes)
  OR EXISTS(special_conditions)
  OR EXISTS(superseded_entry);
END_ENTITY;
(*
```

4.2.23.1 available_from

The available_from specifies a Contract_submission, a Reference_document, or Source_location that the Data_defintion_entry_item information can be obtained from. Available_from need not be specified for a particular Data_definition_entry_item. There may be more than one available_from for a Data_-definition_entry_item.

NOTE Application object EXPRESS:

```
*)
TYPE source_identification = SELECT
  (contract_submission,
  reference_document,
  source_location);
END_TYPE;
(*
```

4.2.23.2 delivery_accounting_references

The delivery_accounting_references specifies the rationale and a reference identification for the data exchange. The rationale and identification provide traceability for configuration control of the collection of data being exchanged. Delivery_accounting_references need not be specified for a particular Data_-definition_entry_item. There may be more than one delivery_accounting_references for a Data_-definition_entry_item.

EXAMPLE 1 A letter of transmittal is a delivery_accounting_reference.

EXAMPLE 2 A material receiving inspection form is a delivery_accounting_reference.

4.2.23.3 entry_files

The entry_files specifies an Exchange_file or Accessed_file for the Data_defintion_entry_item. Entry_files need not be specified for a particular Data_definition_entry_item. There may be more than one entry_files for a Data_definition_entry_item.

EXAMPLE File “ABC.WPD” is at node 124.27.6.4 on the Internet.

NOTE Application object EXPRESS:

```
*)
TYPE data_definition_file_entry = SELECT
  (accessed_file,
  exchange_file);
END_TYPE;
(*
```

4.2.23.4 entry_item_change_level

The entry_item_change_level specifies the revision, change, or issue level that the Data_definition_entry_-item was modified. The order in the listing is chronological. Entry_item_change_level need not be specified for a particular Data_definition_entry_item. There may be more than one entry_item_change_level for a Data_definition_entry_item.

NOTE The entry_item_change_level would not be utilized for the first use or original collection of data in a particular Data_definition_exchange.

4.2.23.5 entry_notes

The entry_notes specify textual information of interest relative to the Data_defintion_entry_item. The textual information is an explanatory remark or notation regarding the Data_definition_entry_item. The entry_notes need not be specified for a particular Data_definition_entry_item. There may be more than one entry_notes for a Data_definition_entry_item.

4.2.23.6 special_conditions

The special_conditions specify mutually agreed to conditions that apply to the Data_defintion_entry_item. The mutually agreed to conditions relate to codes or designations for Files, or Items. Special_conditions need not be specified for a particular Data_definition_entry_item. There may be more than one special_-conditions for a Data_definition_entry_item.

4.2.23.7 superseded_entry

The superseded_entry specifies the identification of an Item that was superseded by the current Data_-definition_entry_item. Superseded_entry need not be specified for a particular Data_definition_entry_item.

EXAMPLE part BE7823-5 supersedes part BE7823-1.

4.2.24 Data_definition_entry_tdp_element

The Data_definition_entry_tdp_element is a collection of characteristics about a Tdp_element (see 4.2.170) in a data exchange.

NOTE 1 The collection of characteristics are typical characteristics about the data that is maintained within Product Data Management or Configuration Control systems. The characteristics are employed to manage the use and distribution of the File, Item, or Tdp_element.

The data associated with a Data_definition_entry_tdp_element are the following:

— actual_inclusion_in_data_exchange;

— available_from;

— data_usage_rights;

— delivery_accounting_references;

— entry_content_property;

— entry_files;

— entry_format;

— entry_item_change_level;

— entry_notes;

— master_file;

— size;

— special_conditions;

— superseded_entry.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY data_definition_entry_tdp_element;
  actual_inclusion_in_data_exchange : OPTIONAL label;
  available_from : OPTIONAL LIST [1:?] OF source_identification;
  data_usage_rights : OPTIONAL label;
  delivery_accounting_references :
      OPTIONAL SET [1:?] OF delivery_accounting;
  entry_content_property : OPTIONAL content_property;
  entry_files : OPTIONAL LIST [1:?] OF data_definition_file_entry;
  entry_format : OPTIONAL entry_format_select;
  entry_item_change_level : OPTIONAL LIST [1:?] OF identifier;
  entry_notes : OPTIONAL LIST [1:?] OF notation;
```



```
    master_file : OPTIONAL yes_no;
    size : OPTIONAL size_characteristics;
    special_conditions : OPTIONAL LIST [1:?] OF special_condition;
    superseded_entry : OPTIONAL element_identification;
  WHERE
    WR1: EXISTS(actual_inclusion_in_data_exchange)
    OR EXISTS(available_from) OR EXISTS(data_usage_rights) OR
    EXISTS(delivery_accounting_references) OR
    EXISTS(entry_content_property) OR EXISTS(entry_files)
    OR EXISTS(entry_format) OR EXISTS(entry_item_change_level)
    OR EXISTS(entry_notes) OR EXISTS(master_file) OR EXISTS(size)
    OR EXISTS(special_conditions) OR EXISTS(superseded_entry);
  END_ENTITY;
  (*
```

4.2.24.1 actual_inclusion_in_data_exchange

The actual_inclusion_in_data_exchange specifies whether or not the File, or Tdp_element is included or furnished as a part of this exchange. The value of the actual_inclusion_in_data_exchange shall be a user defined text string or one of the following:

— already delivered;

— available for access;

— fully included;

— not deliverable;

— not delivered yet;

— not received;

— partially included;

— receipt accepted;

— receipt acknowledged;

— receipt partially rejected;

— receipt rejected;

— resubmission.

NOTE See 4.2.24.1.1 - 4.2.24.1.12, 4.2.49.2.12 for the definition of each allowable value of actual_inclusion_in_-data_exchange.

4.2.24.1.1 already delivered

Already delivered specifies the Tdp_element data is not in the exchange because it was exchanged previously.

4.2.24.1.2 available for access
Available for access specifies the Tdp_element data is available for the receiving system to access remotely the data. The Tdp_element data is not in the exchange.

4.2.24.1.3 fully included
Fully included specifies that the Tdp_element data and all of data files are in the exchange.

4.2.24.1.4 not deliverable
Not deliverable specifies that the Tdp_element data are not in the exchange. The Tdp_element data are not deliverable or planned for delivery.

4.2.24.1.5 not delivered yet
Not delivered yet specifies that the Tdp_element data are not in the current exchange. The data will be exchanged in a future exchange.

4.2.24.1.6 not received
Not received specifies that the Tdp_element data were not received in a referenced exchange or shipment.

4.2.24.1.7 partially included
Partially included specifies that components of the Tdp_element data are exchanged. Some of the data files comprising the Tdp_element are exchanged.

4.2.24.1.8 receipt_accepted
Receipt accepted specifies that the Tdp_element data received in the referenced exchange were successfully received and accepted as approved by the receiver.

4.2.24.1.9 receipt acknowledgement
Receipt acknowledgement specifies that the Tdp_element data received in the referenced exchange or shipment were successfully received.

NOTE The receipt acknowledgement does not imply acceptance by the receiver.

4.2.24.1.10 receipt partially rejected
Receipt partially rejected specifies that the Tdp_element data received in the referenced exchange or shipment were partially rejected due to non-compliance with exchange or delivery requirements.

4.2.24.1.11 receipt rejected
Receipt rejected specifies that the Tdp_element data received in the referenced exchange or shipment were rejected due to non-compliance with the exchange or delivery requirements.

4.2.24.1.12 resubmission
Resubmission specifies that the Tdp_element data are in the exchange as a resubmission or resend.

4.2.24.2 available_from

The available_from specifies a Contract_submission, a Reference_document, or Source_location that the Data_defintion_entry_tdp_element information can be obtained from. Available_from need not be specified

for a particular Data_definition_entry_tdp_element. There may be more than one available_from for a Data_definition_entry_tdp_element.

EXAMPLE When documents available from the International Standards Organization (ISO) are utilized in the preparation of a TDP, the documents are not delivered but are referenced for the receiving organization to acquire through ISO.

NOTE Application object EXPRESS:

```
TYPE source_identification = SELECT
  (contract_submission,
  reference_document,
  source_location);
END_TYPE;
```

4.2.24.3 data_usage_rights

The data_usage_rights specifies the user of the data designated in the Data_definition_entry_tdp_element has legal rights to use the data. The data_usage_rights need not be specified for a particular Data_-definition_entry_tdp_element.

NOTE The source design activity or source system responsible for development of the data of interest may have certain legal rights to limit the use of the data by the activity that is receiving the data of interest. The data_usage_-rights designates the limit of these rights.

EXAMPLE The source design activity responsible for the development of the File, Item, or Tdp_element specifies that the rights of the destination user are stipulated to be “Limited”.

4.2.24.4 delivery_accounting_references

The delivery_accounting_references specifies the rationale and a reference identification for the data exchange. The rationale and identification provide traceability for configuration control of the collection of data being exchanged. Delivery_accounting_references need not be specified for a particular Data_ definition_entry_tdp_element. There may be more than one delivery_accounting_references for a Data_-definition_entry_tdp_element.

EXAMPLE 1 A letter of transmittal is a delivery_accounting_reference.

EXAMPLE 2 A material receiving inspection form is a delivery_accounting_reference.

4.2.24.5 entry_content_property

The entry_content_property specifies characteristics about the the kind of information that is being captured in the Tdp_element in the Data_defintion_entry_tdp_element. Entry_content_property need not be specified for a particular Data_definition_entry_tdp_element.

4.2.24.6 entry_files

The entry_files specifies an Exchange_file or Accessed_file for the Data_defintion_entry_tdp_element. Entry_files need not be specified for a particular Data_definition_entry_tdp_element. There may be more than one entry_files for a Data_definition_entry_tdp_element.

EXAMPLE File “ABC.WPD” is at node 124.27.6.4 on the Internet.

NOTE Application object EXPRESS:

```
TYPE data_definition_file_entry = SELECT
  (accessed_file,
  exchange_file);
END_TYPE;
```

4.2.24.7 entry_format

The entry_format specifies a File_format or a human-interpretable string of characters that specifies the digital file format or the format for the physical hardcopy media that is used to exchange the data designated in the Data_definition_entry_tdp_element. The entry_format need not be specified for a particular Data_-definition_entry_tdp_element.

NOTE 1 The digital file format is independent of the digital media that the file is transferred on.

NOTE 2 Application object EXPRESS:

```
*)
TYPE entry_format_select = SELECT
  (file_format,
  text);
END_TYPE;
(*
```

EXAMPLE 1 Physical hardcopy media is paper, mylar, film, or aperture cards.

EXAMPLE 2 Digital media is 9 track tape, floppy disk, diskette, or 8mm cartridge tape.

4.2.24.8 entry_item_change_level

The entry_item_change_level specifies the revision, change, or issue level that the Data_definition_entry_-tdp_element was modified. The order in the listing is chronological. Entry_item_change_level need not be specified for a particular Data_definition_entry_tdp_element. There may be more than one entry_item_-change_level for a Data_definition_entry_tdp_element.

NOTE The entry_item_change_level would not be utilized for the first use or original collection of data in a particular Data_definition_exchange.

4.2.24.9 entry_notes

The entry_notes specify textual information of interest relative to the Data_defintion_entry_tdp_element. The textual information is an explanatory remark or notation regarding the Data_definition_entry_tdp_-element. The entry_notes need not be specified for a particular Data_definition_entry_tdp_element. There may be more than one entry_notes for a Data_definition_entry_tdp_element.

4.2.24.10 master_file

The master_file specifies whether or not the Data_definition_entry_tdp_element is considered the master representation for legal purposes. Master_file need not be specified for a particular Data_definition_entry_-tdp_element.

EXAMPLE A drawing is sent to a user in a raster and in an ISO 10303-202 representation using an ISO 10303-21 file format. The raster file is flagged as the master_file.

NOTE 1 There may be exist multiple digital representations of a File, Item, or Tdp_element. The master_file specifies which representation is used for legal purposes.

NOTE 2 When multiple representations exist of a File, Item, or Tdp_element, then each representation will be recorded as a separate Data_definition_entry_tdp_element.

NOTE 3 The yes_no is a logical flag stating something is, is not, or not known.

NOTE 4 Application object EXPRESS:

```
*)
TYPE yes_no = logical;
END_TYPE;
(*
```

4.2.24.11 size

The size specifies the non-digital information or digital file size for all Files, Items, or Tdp_elements that the Data_definition_entry_tdp_element represents. Size need not be specified for a particular Data_-definition_entry_tdp_element.

EXAMPLE 1 The files size is 63 megabytes.

EXAMPLE 2 The drawing consists of 36 aperture cards.

EXAMPLE 3 The drawing consists of 12 sheets.

4.2.24.12 special_conditions

The special_conditions specify mutually agreed to conditions that apply to the Data_defintion_entry_tdp_-element. The mutually agreed to conditions relate to codes or designations for Files, Items, or Tdp_-elements. Special_conditions need not be specified for a particular Data_definition_entry_tdp_element. There may be more than one special_conditions for a Data_definition_entry_tdp_element.

4.2.24.13 superseded_entry

The superseded_entry specifies the identification of a Tdp_element that was superseded by the current Data_definition_entry_tdp_element. Superseded_entry need not be specified for a particular Data_-definition_entry_tdp_element.

EXAMPLE ISO 10303-203:2001 supersedes ISO 10303-203:1995.

4.2.25 Data_definition_exchange

Data_definition_exchange is a data exchange that is needed to satisfy a particular business purpose. The Data_definition_exchange defines sufficient configuration and data administration information to manage the data exchange. The business purpose may be a specific business data exchange or shipping transactions. The data identified in the Data_definition_exchange is the product definition data. The data associated with Data_definition_exchange are the following:

— list_body;

— list_header;

— presentation.

NOTE 1 The listing in the Data_definition_exchange may be in a “top-down-break-down” order.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY data_definition_exchange;
  list_body : data_definition_exchange_body;
  list_header : data_definition_exchange_header;
  presentation: OPTIONAL list_presentation;
END_ENTITY;
(*
```

4.2.25.1 list_body

The list_body specifies the data tabulations that are the constituents and subject of the Data_definition_-exchange.

4.2.25.2 list_header

The list_header specifies sufficient configuration control and data management information for a specific Data_definition_exchange.

4.2.25.3 presentation

The presentation specifies the human interpretable graphical display of the Data_definition_exchange information. Presentation need not be specified for a particular Data_definition_exchange.

NOTE 1 The presentation of the Data_definition_exchange information is available for human interpretation of the information through a software application that can interpret and display the information for a human.

NOTE 2 In some judicial systems and in many contractual situations, the human interpretation of the digital data in the exchange is considered the binding legal instrument.

4.2.26 Data_definition_exchange_body

The Data_definition_exchange_body is the tabulation of information within the Data_definition_exchange (see 4.2.25). The tabulated information consists of sufficient identification of Items (see 4.2.89), Files (see 4.2.59), or Tdp_elements (see 4.2.170) to allow the receiving user to manage and use the information that is contained in the exchange.

The data associated with a Data_definition_exchange_body are the following:

— list_method;

— notes_list;

— revision_history.

NOTE Application object EXPRESS:

```
*)
ENTITY data_definition_exchange_body;
  list_method : data_definition_exchange_list_method;
  notes_list : OPTIONAL LIST [1:?] OF notation;
  revision_history : OPTIONAL LIST [1:?] OF revision;
END_ENTITY;
(*
```

4.2.26.1 list_method

The list_method specifies a Simple_list_of_elements Simple_list_of_files, or a Data_definition_indentured_-list_method that will used to collect and related the technical data package elements that are contained in Data_definition_exchange.

NOTE 1 The data_definition_indentured_list_method provides an indentured list of technical data package elements, using one of the three indentured methods, with or without their related data files. The three indentured methods are part based, document based, and part based with document references to parts.

NOTE 2 The simple_list_of_elements identifies just a simple list of technical data package elements to be exchanged (with or without their related data files).

NOTE 3 The simple_list_of_files identifies just a simple list of data files to be exchanged.

NOTE 4 Application object EXPRESS:

```
*)
TYPE data_definition_exchange_list_method = SELECT
  (data_definition_indentured_list_method,
  simple_list_of_elements,
  simple_list_of_files);
END_TYPE;
(*
```

4.2.26.2 notes_list

The notes_list specifies notes that are applicable to the Data_definition_exchange. Some notes are referenced by the use of a note reference code in any section of the Data_definition_exchange. The notes describe explanatory remarks or notations that may be useful or informative regarding the subject that referenced the note. The notes_list need not be specified for a particular Data_definition_exchange_body. There may be more than one notes_list for a Data_definition_exchange_body.

4.2.26.3 revision_history

The revision_history specifies a record of revisions of the Data_definition_exchange. The revision_history shall be constructed in a manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_-history shall not contain an entry for the original release. The revision_history need not be specified for a particular Data_definition_exchange_body. There may be more than one revision_history for a Data_-definition_ exchange_body.

4.2.27 Data_definition_exchange_header

The Data_definition_exchange_header is the configuration control, data management, and usage information necessary to manage and control the Data_definition_exchange. The data associated with Data_-definition_exchange_header are the following:

— common_header;

— date_of_transfer;

— delivery_accounting_references;

— destinations;

— exchange_reason;

— procurement_references.

NOTE Application object EXPRESS:

```
*)
ENTITY data_definition_exchange_header;
  common_header : header;
  date_of_transfer : OPTIONAL date;
  delivery_accounting_references : OPTIONAL SET [1:?] OF
    delivery_accounting;
  destinations : OPTIONAL SET [1:?] OF system_destination;
  exchange_reason : reason;
  procurement_references: OPTIONAL LIST [1:?] OF identification_select;
END_ENTITY;
(*
```

4.2.27.1 common_header

The common_header specifies the information required for administering the configuration control and use of the data in the Data_definition_exchange.

4.2.27.2 date_of_transfer

The date_of_transfer specifies the date and time when the responsible activity initiated the transfer of the Data_definition_exchange. The date_of_transfer need not be specified for a particular Data_definition_-exchange_header.

4.2.27.3 delivery_accounting_references

The delivery_accounting_references specify the information required for the business rationale for the delivery of the Data_definition_exchange information. The delivery_accounting_references need not be specified for a particular Data_definition_exchange_header. There may be more than one delivery_-accounting_references for a Data_definition_exchange_header.

EXAMPLE Delivery accounting information is letters of transmittals, Purchase Orders, and Material Inspection Report Forms.

4.2.27.4 destinations

The destinations specify the target organizations where the Data_definition exchange is transmitted. The destinations need not be specified for a particular Data_definition_exchange_header. There may be more than one destinations for a Data_definition_exchange_header.

4.2.27.5 exchange_reason

The exchange_reason specifies the purpose for the Data_definition_exchange information exchange.

4.2.27.6 procurement_references

The procurement_references specifies an Item_identification or an Element_identification that are Tdp_-elements, documents, Files, or Items that were used as justification for development of the Data_definition_-

exchange. The justification is a business purpose that identifies a Tdp_element, document, File, or Item that was provided by the organization procuring the data contained in the Data_definition_exchange. The business purpose permits the association of different Data_definition_exchange files in a hierarchial or other business purpose defined method. The procurement_references need not be specified for a particular Data_-definition_exchange_header. There may be more than one procurement_references for a Data_definition_-exchange_header.

NOTE Application object EXPRESS:

```
*)
TYPE identification_select = SELECT
  (element_identification,
  item_identification);
END_TYPE;
(*
```

4.2.28 Data_definition_exchange_simple_entry

The Data_definition_exchange_simple_entry is an association between an identification of a Tdp_ element (see 4.2.170) and the exchange parameters of the Tdp_element. The association is defined by the business purpose of the exchange. The data associated with a Data_definition_exchange_simple_entry are the following:

— entry;

— simple_entry.

NOTE Application object EXPRESS:

```
*)
ENTITY data_definition_exchange_simple_entry;
  entry : data_definition_entry_tdp_element;
  simple_entry : element_identification;
END_ENTITY;
(*
```

4.2.28.1 entry

The entry specifies the characteristics about the simple_entry for the Data_definition_exchange_simple_-entry.

4.2.28.2 simple_entry

The simple_entry specifies the Tdp_element of interest for the Data_definition_exchange_simple_entry.

4.2.29 Data_definition_indentured_entry

The Data_definition_indentured_entry is an association between either the Tdp_element (see 4.2.170) or Item (see 4.2.89) in the indentured list and their entry characteristics. The association is a one-to-one association between either the Tdp_element or Item (see 4.2.89) and the characteristics about the Tdp_-

element orItem. The Tdp_element or Item of interest defines a relationship of either the Tdp_element or Item to the next higher assembly in the relationship tree, the next higher Tdp_element in the relationship tree, the top level Tdp_element, or the top level Item.

NOTE 1 The type of relationship tree (for example, drawing, part) is defined in the Data_definition_indentured_list_method.

The data associated with a Data_definition_exchange_indentured_entry are the following:

— effective_on;

— entry_characteristics;

— indenture_level;

— indentured_entry.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY data_definition_indentured_entry;
  effective_on : OPTIONAL effectivity;
  entry_characteristics : data_definition_entry_select;
  indentured_entry : list_entry_select;
  indenture_level: OPTIONAL label;
END_ENTITY;
(*
```

4.2.29.1 effective_on

The effective_on specifies parameters that governs the use of the Item. The effective_on need not be specified for a particular data_definition_indentured_entry.

4.2.29.2 entry_characteristics

The entry_characteristics specifies the aspects about the indentured_entry for the Data_definition_-indentured_entry.

NOTE 1 There is a different set of allowable entry characteristics for an Item and a Tdp_element

NOTE 2 Application object EXPRESS:

```
*)
TYPE data_definition_entry_select = SELECT
  (data_definition_entry_item,
   data_definition_entry_tdp_element);
END_TYPE;
(*
```

4.2.29.3 indenture_level

The indenture_level indicates the number of indents for the Data_definition_indentured_entry from the Item or Tdp_element identified in the indentured_entry. The level of indenture indicates a relative position between the top level Item, or Tdp_element and the Data_definition_indentured_entry that is specified in the indentured_entry. The indenture_level need not be specified for a particular Data_definition_indentured_-entry.

NOTE 1 The level of indenture is formulated based on the document tree or the item tree from the top level Tdp_-element or Item identified in the data_definition_indentured_entry.indentured_entry.

NOTE 2 Industrial practices have the Tdp_element or Item as the first level of indenture, while the Tdp_elements or Items that are referenced from the top level Tdp_element or Item are at the second level of indenture. Lower levels of indenture are specified relative to the top level Tdp_element or Item.

4.2.29.4 indentured_entry

The indentured_entry specifies an Item_parent_to_item_child_relationship, a Tdp_element_parent_to_item_-child_relationship, a Tdp_element_parent_to_tdp_element_child_relationship, Tdp_indentured_tdp_element, an Item_parent_to_tdp_element_child_relationship, or a Tdp_indentured_item for the Data_definition_-indentured_entry and the relationship of the Tdp_element or item to other Tdp_ elements or items.

NOTE 1 An Item_parent_to_item_child_relationship specifies that the row entry (child) is an Item and that the parent entry is an Item.

NOTE 2 A Tdp_element_parent_to_item_child_relationship specifies that the row entry (child) is an Item and that the parent entry is a Tdp_element.

NOTE 3 A Tdp_element_parent_to_tdp_element_child_relationship specifies that the row entry (child) is a TDP_-element and that the parent entry is a Tdp_element.

NOTE 4 The Tdp_indentured_tdp_element specifies the top element of a Tdp_element based tree. This enables the listing of multiple document trees within the body of the list.

NOTE 5 An Item_parent_to_tdp_element_child_relationship specifies that the row entry (child) is a Tdp_element and that its parent entry is an Item.

NOTE 6 The Tdp_indentured_item specifies the top Items of an Item based tree. This enables the listing of multiple part-based document trees within the body of the list.

NOTE 7 Application object EXPRESS:

```
*)
TYPE list_entry_select = SELECT
  (item_parent_to_item_child_relationship,
   item_parent_to_tdp_element_child_relationship,
   tdp_element_parent_to_item_child_relationship,
   tdp_element_parent_to_tdp_element_child_relationship,
```



```
   tdp_indentured_item,
   tdp_indentured_tdp_element);
END_TYPE;
(*
```

4.2.30 Data_definition_indentured_list_method

The data_definition_indentured_list_method is an ordered collection of Tdp_elements or Items defined by one of three indentured methods. The data associated with a Data_definition_indentured_list_method are the following:

— method_of_entry_tabulation;

— order_of.

NOTE Application object EXPRESS:

```
*)
ENTITY data_definition_indentured_list_method;
  method_of_entry_tabulation : indentured_list_method;
  order_of : LIST [1:?] OF data_definition_entry_and_file_select;
END_ENTITY;
(*
```

4.2.30.1 method_of_entry_tabulation

The method_of_entry_tabulation specifies an Indentured_list_by_part, an Indentured_list_by_document, or an Indentured_list_by_part_with_document_references_to_parts that specify how the entries in the order_of are specified for the Data_definition_exchange, or Indentured_data_definition_list.

NOTE The three methods are defined by the Indentured_list_by_part, the Indentured_list_by_part_with_-document_references_to_parts, or the Indentured_list_by_document.

4.2.30.2 order_of

The order_of specifies the list of Data_definition_indentured_entry(s) and data_definition_file_entry(s) for the Data_definition_exchange. The order in the listing is specified by the method_of_entry_tabulation. There may be more than one order_of for a data_definition_indentured_list_method.

NOTE Application object EXPRESS:

```
*)
TYPE data_definition_entry_and_file_select = SELECT
  (data_definition_indentured_entry,
  data_definition_file_entry);
END_TYPE;
(*
```

4.2.31 Data_list

A Data_list is a type of Single_document_list (see 4.2.148) that contains a tabulation of all drawings, reference documents, subordinate data lists, associated lists, and other essential in-house documents required for design disclosure pertaining to the Tdp_element (see 4.2.170) or Item (see 4.2.89) to which the Data_list applies. The data associated with a Data_list are the following:

— body;

— list_header.

NOTE Application object EXPRESS:

```
*)
ENTITY data_list
  SUBTYPE OF (single_document_list);
  body : data_list_body;
  list_header : data_list_header;
END_ENTITY;
(*
```

4.2.31.1 body

The body specifies the tabulations that are the constituents and subject of the Data_list.

4.2.31.2 list_header

The list_header specifies sufficient configuration control and data management information for a specific Data_list.

4.2.32 Data_list_body

The Data_list_body is the tabulation of information within the Data_list (see 4.2.31). The tabulated information consists of sufficient identification of related Tdp_element (see 4.2.170) information to meet design disclosure requirements for the Tdp_ element that is associated to the Data_list. The related Tdp_-element information is identified within the context of the Tdp_element or item that is associated to the Data_list. The data associated with a Data_list_body are the following:

— data_list_notes;

— data_list_tabulations;

— revision_history.

NOTE Application object EXPRESS:

```
*)
ENTITY data_list_body;
  data_list_notes : OPTIONAL LIST [1:?] OF notation;
  data_list_tabulations : LIST [1:?] OF data_list_tabulation;
```



```
  revision_history : OPTIONAL LIST [1:?] OF revision;
END_ENTITY;
(*
```

4.2.32.1 data_list_notes

The data_list_notes specifies notes that are applicable to the Data_list. Some notes are referenced by the use of a note reference code in any section of the Data_list. The notes describe any explanatory remarks or notations that may be useful or informative regarding the subject that referenced the note. The data_list_-notes need not be specified for a particular Data_list_body. There may be more than one data_list_notes for a Data_list_body.

4.2.32.2 data_list_tabulations

The data_list tabulations specify a list of valid Tdp_element that are contained within the Data_list. There may be more than one data_list_tabulations for a Data_list_body.

NOTE When several tabulations are required, each tabulation is for a separate item.

4.2.32.3 revision_history

The revision_history specifies a record of revisions of the Data_list. The revision_history shall be constructed in a manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_history shall not contain an entry for the original release. The revision_history need not be specified for a particular Data_list_body. There may be more than one revision_history for a Data_list_body.

4.2.33 Data_list_entry

A Data_list_entry is an entry in a Data_list_tabulation (see 4.2.35). The data associated with a Data_list_-entry are the following:

— available_from;

— entry_item_change_level;

— entry_notes;

— list_entry;

— special_conditions.

NOTE Application object EXPRESS:

```
*)
ENTITY data_list_entry;
     available_from: OPTIONAL LIST [1:?] OF source_identification;
     entry_item_change_level : OPTIONAL identifier;
     entry_notes : OPTIONAL LIST [1:?] OF notation;
```

```
        list_entry : valid_tdp_elements_for_data_list;
        special_conditions: OPTIONAL LIST [1:?] OF special_condition;
    END_ENTITY;
    (*
```

4.2.33.1 available_from

The available_from specifies a Contract_submission, a Reference_document, or Source_location where a list_entry is available. Available_from need not be specified for a particular Data_list_entry. There may be more than one available_from for a Data_list_entry.

NOTE 1 Available_from identifies a company or organization that can provide the list_entry Tdp_element.

NOTE 2 In lieu of providing Tdp_elements that are used widely in Industrial practices, a design activity may identify where a Tdp_element can be obtained.

NOTE 3 Application object EXPRESS:

```
TYPE source_identification = SELECT
  (contract_submission,
  reference_document,
  source_location);
END_TYPE;
```

EXAMPLE A Tdp_element is ISO 9000. The Tdp_element is available_from the International Standards Organization (ISO).

4.2.33.2 entry_item_change_level

The entry_item_change_level specifies the current change level of the Data_list_entry. The entry_item_-change_level need not be specified for a particular Data_list_entry.

NOTE 1 Industrial practices utilize entry_item_change_level to notify users of a change to the line item of interest.

NOTE 2 Industrial practices identify the last change of the respective entry.

4.2.33.3 entry_notes

The entry_notes specifies notes that are applicable to the Data_list_entry. The entry_notes need not be specified for a particular Data_list_entry. There may be more than one entry_notes for a Data_list_entry.

4.2.33.4 list_entry

The list_entry specifies a Reference_document, Header, or a Sheet that specifies the allowable Tdp_element of interest.

NOTE 1 Header provides a general identification for Tdp_elements such as Data_list, Drawing,Parts_list, Other_ list, and Product_data_set.

NOTE 2 Application object EXPRESS:

```
*)
TYPE valid_tdp_elements_for_data_list = SELECT
     (header,
     sheet,
     reference_document);
END_TYPE;
(*
```

4.2.33.5 special_conditions

The special_conditions specify mutually agreeable conditions that apply to the Data_list_entry. The mutually agreed to conditions relate to codes or designations for the Tdp_element. Special_conditions need not be specified for a particular Data_list_entry. There may be more than one special_conditions for a Data_list_entry.

4.2.34 Data_list_header

A Data_list_header is the configuration control and data management information necessary to manage and control the Data_list (see 4.2.31) for an Item (see 4.2.89) or a Tdp_element (see 4.2.170). The data associated with a Data_list_header are the following:

— common_header.

NOTE Application object EXPRESS:

```
*)
ENTITY data_list_header;
  common_header : header;
END_ENTITY;
(*
```

4.2.34.1 common_header

The common_header specifies the information required for configuration control and administering the use of the data in the Data_list.

4.2.35 Data_list_tabulation

The Data_list_tabulation is a listing of valid Tdp_elements for the associated Tdp_element (see 4.2.170). The listing identifies the valid Tdp_elements for the Item (see 4.2.89) associated to the Tdp_element. The data associated with a Data_list_tabulation are the following:

— data_list_entries;

— for_item.

NOTE Application object EXPRESS:

```
*)
ENTITY data_list_tabulation;
  data_list_entries : OPTIONAL LIST [1:?] OF data_list_entry;
  for_item : OPTIONAL item_identification;
WHERE
  WR1: EXISTS (data_list_entries) OR EXISTS (for_item);
END_ENTITY;
(*
```

4.2.35.1 data_list_entries

The data_list_entries specifies a listing of the Tdp_elements that are required for design disclosure of the Tdp_element that the Data_list is associated to. When a for_item is specified, the listing is for the item designated in the for_item. Data_list_entries need not be specified for a particular Data_list_tabulation. There may be more than one data_list_entries for a Data_list_tabulation.

4.2.35.2 for_item

The for_item specifies the Item associated to the data_list_entries. For_item need not be specified for a particular data_list_tabulation.

NOTE 1 For industrial practices of one Item for one Drawing (for example., one part - one drawing), the for_item is assumed to be the Item defined on the drawing.

NOTE 2 For industrial practices of multiple Items defined on a drawing, the for_item may be different for each Item.

4.2.36 Date

A Date is the identification of the day, month, and year in history. The data associated with Date are the following:

— day;

— month;

— specific_time;

— week;

— year.

NOTE Application object EXPRESS:

```
*)
ENTITY date;
  day: OPTIONAL an_integer;
  month: OPTIONAL an_integer;
  specific_time: OPTIONAL time;
```



```
  week: OPTIONAL an_integer;
  year: an_integer;
END_ENTITY;
(*
```

4.2.36.1 day

The day specifies a specific period of time that the earth turn once on its axis. Approximately 365.25 days are in a year. The day need not be specified for a particular Date.

4.2.36.2 month

The month specifies a specific period of time corresponding approximately to 28 days or four weeks or the divided 12 periods of a year. The month need not be specified for a particular Date.

4.2.36.3 specific_time

A specific_time specifies a moment of occurrence measured by hour, minute, and second. The specific_time need not be specified for a particular Date.

4.2.36.4 week

The week specifies a specific period of time corresponding to 7 days. Approximately 52 weeks are in a year. The week need not be specified for a particular Date.

4.2.36.5 year

A year specifies the year as defined in the Gregorian Calendar.

NOTE ISO 8601 defines the Gregorian Calendar.

4.2.37 Date_effectivity

The Date_effectivity is a type of Effectivity (see 4.2.45) that specifies the expected or actual usage date of a component in a Product_model (see 4.2.127).

NOTE 1 The preceding definition is a broader definition that is defined for ISO 10303-203 for Planned_date_-effectivity.

The data associated with a Date_effectivity are the following:

— end_date;

— start_date.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY date_effectivity
  SUBTYPE OF (effectivity);
  end_date : OPTIONAL date;
  start_date : date;
END_ENTITY;
(*
```

4.2.37.1 end_date

The end_date specifies the last date that the component is to be used or was used in the Product_-configuration. No end_date at the time of exchange signifies that an end_date has not yet been defined. The end_date need not be specified for a particular Date_effectivity.

NOTE The preceding definition is a broader definition that is defined for ISO 10303-203 for Planned_effectivity end_date.

4.2.37.2 start_date

The start_date specifies the first date that the component is to be used or was used in the Product_-configuration.

NOTE The preceding definition is a broader definition that is defined for ISO 10303-203 for Planned_date_effectivity. start_date.

4.2.38 Delivery_accounting

The Delivery_accounting is delivery information of the Item (see 4.2.89), File (see 4.2.59), or Tdp_element (see 4.2.170). Delivery information identifies and describes the supporting business rationale for the delivery of the Item, File, Tdp_element, or document of interest. The data associated with a Delivery_accounting are the following:

— reference_identification;

— reference_identification_description.

NOTE Application object EXPRESS:

```
*)
ENTITY delivery_accounting;
  reference_identification : identifier;
  reference_identification_description : text;
END_ENTITY;
(*
```


4.2.38.1 reference_identification

The reference_identification specifies the delivery accounting instrument identification. The delivery accounting instrument is the business purpose of the data delivery.

4.2.38.2 reference_identification_description

The reference_identification_description specifies the title of the delivery accounting instrument.

EXAMPLE Letters of transmittals, Purchase Orders, and Material Inspection Report Forms.

4.2.39 Design_authority

The Design_authority is the organization, company, business or industry responsible for a design. The data associated with a Design_authority are the following:

— design_activity_code;

— design_activity_identification;

— person_responsible;

— role.

NOTE Application object EXPRESS:

```
*)
ENTITY design_authority;
  design_activity_code : OPTIONAL company_code;
  design_activity_identification : company;
  person_responsible : OPTIONAL person;
  role : OPTIONAL label;
END_ENTITY;
(*
```

4.2.39.1 design_activity_code

The design_activity_code specifies a company identifier code for the Design_authority responsible for the design content. The design_activity_code need not be specified for a particular Design_authority.

NOTE Many industry associations and governments assign unique design_activity_codes to each company.

EXAMPLE Dunn & Bradstreet assigns a unique identifier to a requesting organization that identifies the corporation, division (if required), and organization (if required) called the DUNS number.

4.2.39.2 design_activity_identification

The design_activity_identification specifies the company responsible for the design content.

4.2.39.3 person_responsible

The person_responsible specifies the person responsible for the development of the design content within the Design_authority. The person_responsible need not be specified for a particular Design_authority.

4.2.39.4 role

The role specifies the purpose of the Design_authority. The role need not be specified for a particular Design_authority.

4.2.40 Distribution_notice

A Distribution_notice is the allowable industrial or organizational circulation of information for the organization receiving the Item (see 4.2.89), File (see 4.2.59), Tdp_element (see 4.2.170), or document.

The data associated with a Distribution_notice are the following:

— distribution_authority;

— distribution_code;

— distribution_statement.

EXAMPLE A distribution notice indicating `Distribution is Unlimited`

NOTE Application object EXPRESS:

```
*)
ENTITY distribution_notice;
  distribution_authority : person_and_organization;
  distribution_code : OPTIONAL label;
  distribution_statement : OPTIONAL text;
WHERE
  WR1: EXISTS (distribution_statement) OR
  EXISTS (distribution_authority) OR EXISTS (distribution_code);
END_ENTITY;
(*
```

4.2.40.1 distribution_authority

The distribution_authority specifies the Distribution_notice controlling authority for circulation.

4.2.40.2 distribution_code

The distribution_code specifies the Distribution_notice code that identifies the allowed distribution. The distribution_code need not be specified for a particular Distribution_notice.

NOTE Many companies and industries agree on different codes to depict different conditions for circulation.

EXAMPLE Distribution Code “A” indicates “Distribution is Unlimited.”

4.2.40.3 distribution_statement

The distribution_statement specifies the Distribution_notice statement of allowed circulation. The distribution_statement need not be specified for a particular Distribution_notice.

EXAMPLE "Distribution is Unlimited" is a Distribution_statement.

4.2.41 Document_list

A Document_list is a type of Associated_list (see 4.2.11) that specifies all documents, or Tdp_elements (see 4.2.170) that are required to fully document the design disclosure. Each Document_list is either an Indentured_data_list (see 4.2.74) or a Single_document_list (see 4.2.148).

NOTE Application object EXPRESS:

```
*)
ENTITY document_list
  ABSTRACT
  SUPERTYPE OF (ONEOF(indentured_data_list,single_document_list))
  SUBTYPE OF (associated_list);
END_ENTITY;
(*
```

4.2.42 Document_usage_parameter

A Document_usage_parameter is the identification and description of a specific piece of information that is contained in a document. This provides a general capability to define parameters that can be applied to an Item (see 4.2.89).

The data associated with a Document_usage_parameter are the following:

— subject;

— value_of.

EXAMPLE A material code with its value.

NOTE Application object EXPRESS:

```
*)
ENTITY document_usage_parameter;
  subject : label;
  value_of : text;
END_ENTITY;
(*
```

4.2.42.1 subject

The subject specifies the nomenclature for the value_of of the parameter.

4.2.42.2 value_of

The value_of specifies the specific parameter.

4.2.43 Drawing

A Drawing is a type of Tdp_element (see 4.2.170) that is the presentation of product data in a human interpretable form wherein the physical and functional requirements for that product are presented pictorially and textually.

The data associated with a Drawing are the following:

— heading;

— pages.

NOTE Application object EXPRESS:

```
*)
ENTITY drawing
  SUBTYPE OF (tdp_element);
  heading : header;
  pages : LIST [1:?] OF sheet;
END_ENTITY;
(*
```

4.2.43.1 heading

The heading specifies the identification and configuration control information for the Drawing.

4.2.43.2 pages

The pages specifies the listing of sheets of a Drawing. The order of the listing is based on the sheet number of the respective sheet in the Drawing. There may be more than one pages for a drawing.

4.2.44 Drawing_suffix_number_combination

The Drawing_suffix_number_combination is the identifying number of an Item (see 4.2.89) when the Item is identified using drawing practices of combining the drawing number and a suffix on the face of the drawing. The data associated with an Drawing_suffix_number_combination are the following:

— drawing_number;

— suffix_identifier.

NOTE Application object EXPRESS:

```
*)
ENTITY drawing_suffix_number_combination;
  drawing_number : element_identification;
  suffix_identifier : identifier;
END_ENTITY;
(*
```

4.2.44.1 drawing_number

The drawing_number specifies the identification of the Drawing that defines the Item.

4.2.44.2 suffix_identifier

The suffix_identifier specifies the suffix that is associated with the Item on the face of the drawing.

4.2.45 Effectivity

An Effectivity is the intended use of a component in a particular configuration of a product. Each Effectivity may be one of the following: a Sequence_effectivity (see 4.2.142), a Date_effectivity (see 4.2.37), or a Lot_effectivity (see 4.2.103). The data associated with an Effectivity are the following:

— description;

— effectivity_approvals;

— name;

— product;

— work_activity.

NOTE Application object EXPRESS:

```
*)
ENTITY effectivity
  SUPERTYPE OF (ONEOF(sequence_effectivity,lot_effectivity,
      date_effectivity, time_interval_effectivity));
  description: OPTIONAL text;
  effectivity_approvals : OPTIONAL LIST [1:?] OF approval;
  name : OPTIONAL label;
  product : product_configuration;
  work_activity : OPTIONAL label;
END_ENTITY;
(*
```

4.2.45.1 description

The description specifies the characterization of the Effectivity when the Effectivity is not defined by either: a Sequence_effectivity, a Lot_effectivity, or a Date_effectivity. The description need not be specified for a particular Effectivity.

4.2.45.2 effectivity_approvals

The effectivity_approvals specifies the valid approvals that authorize the use of the product for the required effectivity. There may be more than one effectivity_approvals for an Effectivity. The effectivity_approvals need not be specified for a particular Effectivity.

NOTE The approvals are determined by the Design_authority.

4.2.45.3 name

The name specifies the words by which the Effectivity is known. The name need not be specified for a particular Effectivity.

NOTE The name could be used to assist in distinguishing between configuration effectivity and product definition effectivity

4.2.45.4 product

The product specifies the physical unit that is valid for the Effectivity.

4.2.45.5 work_activity

The work_activity specifies the activity that applied the effectivity to some product data. The work_activity need not be specified for a particular Effectivity.

EXAMPLE Identifying the activity that applied a manufacturing effectivity to some product data.

4.2.46 Element_identification

The Element_identification is configuration control identification information for a specific group of product information of interest.

NOTE 1 A specific group of product information may be a Tdp_element, or an other product information grouping defined outside this standard. A specific group of product information may be configuration controlled as a paper document, a single file, multiple files, or within a data base.

The data associated with an Element_identification are the following:

— alternate_identifications;

— change_status;

— design_activities;

— element_certification;

— element_classifications;

— identifying_number;

— outstanding_changes;

— title.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY element_identification;
  alternate_identifications : OPTIONAL SET [1:?] OF
     alternate_identification_element_select;
  change_status : OPTIONAL change_identification;
  design_activities : SET [1:?] OF design_authority;
  element_certification : OPTIONAL certification;
  element_classifications: LIST [1:?] OF element_type;
  identifying_number : identifier;
  outstanding_changes : OPTIONAL LIST [1:?] OF other_reference_document;
  title : OPTIONAL text;
END_ENTITY;
(*
```

4.2.46.1 alternate_identifications

The alternate_identifications specify an Alternate_element_identification or an Identifier that provide additional ways that a Tdp_element may be identified. The alternate_identifications need not be specified for a particular Element_identification. There may be more than one alternate_identifications for an Element_identification.

EXAMPLE 1 If a company wants to rename a document they received from a supplier use Alternate_element_-identification option.

EXAMPLE 2 If a company wants to alias the id of a document because the character string is too long use Identifier option.

NOTE Application object EXPRESS:

```
*)
TYPE alternate_identification_element_select = SELECT
  (alternate_element_identification,
   identifier);
END_TYPE;
(*
```

4.2.46.2 change_status

The change_status specifies the identification of the most recent change information that is recorded for the Element_identification of interest. The change_status need not be specified for a particular Element_-identification.

4.2.46.3 design_activities

The design_activities specifies the organization, company, business, or industry responsible for the Element_identification. There may be more than one design_activities for an Element_identification.

4.2.46.4 element_certification

The element_certification specifies that a Tdp_element has fulfilled the requirements identified in the Certification. The element_certification need not be specified for a particular Element_identification.

4.2.46.5 element_classifications

The element_classifications specifies a classification code for the specific group of product information of interest. The classification code is utilized for data management of the specific group of product information of interest. There may be more than one element_classifications for an Element_identification.

EXAMPLE "PL" is an element_classifications for a Parts list.

4.2.46.6 identifying_number

The identifying_number specifies an identification number for the specific group of product information of interest as issued by the Design_activity.

NOTE The identifying_number may be an alphanumeric.

4.2.46.7 outstanding_changes

The outstanding_changes specifies the identification of one or more changes to a specific group of product information of interest that have not been incorporated. The outstanding_changes need not be specified for a particular Element_identification. There may be more than one outstanding_changes for an Element_-identification.

4.2.46.8 title

The title specifies the formal designation of the specific group of product information of interest. The title need not be specified for a particular Element_identification.

NOTE The document title may be an abbreviated form of the title or may have abbreviations of selected words within the title.

4.2.47 Element_type

The Element_type is a classification code utilized to classify the product data contents of an Element_-identification (see 4.2.46). The data associated with an Element_type are the following:

— code_administrator;

— element_code;

— type_of_coding_scheme.

NOTE 1 Element_types are defined within different industries.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY element_type;
   code_administrator : OPTIONAL company;
   element_code : label;
   type_of_coding_scheme : OPTIONAL text;
END_ENTITY;
(*
```

4.2.47.1 code_administrator

The code_administrator specifies the organization that maintains the element type coding schema. The code_administrator need not be specified for a particular Element_type.

4.2.47.2 element_code

The element_code specifies the classification code for a Tdp_element.

4.2.47.3 type_of_coding_scheme

The type_of_coding_scheme specifies industrial standards or a business transaction specific method to classify the Tdp_element by the information content of the Tdp_element. The standards identify a coding scheme by which types of statements can be referenced. The type_of_coding_scheme need not be specified for a particular Element_type.

EXAMPLE ISO 11179-6 defines a coding scheme that an industry may use.

4.2.48 Event

The Event is something that occurs or will occur at a point in time. The data associated with an Event is the following:

— actual_date;

— assignment;

— description;

— id;

— name;

— planned_date;

— responsible_person_organization.

NOTE Application object EXPRESS:

```
*)
ENTITY event;
  actual_date : OPTIONAL date;
  assignment : OPTIONAL project;
  description : OPTIONAL text;
  id : identifier;
  name : label;
  planned_date : OPTIONAL date;
  responsible_person_organization : OPTIONAL person_and_organization;
END_ENTITY;
(*
```

4.2.48.1 actual_date

The actual_date specifies the date and or the time when the Event occured. The actual_date need not be specified for a particular Event.

4.2.48.2 assginment

The assignment specifies the product data that the Event is associated with. The assignment need not be specified for a particular Event.

4.2.48.3 description

The description specifies the text that describes the Event. The description need not be specified for a particular Event.

4.2.48.4 id

The id specifies the identification of the Event assigned by the organization.

4.2.48.5 name

The name specifies the words by which the Event is known.

4.2.48.6 planned_date

The planned_date specifies the date and or the time when the Event may occur. The planned_date need not be specified for a particular Event.

4.2.48.7 responsible_person_organization

The responsible_person_organization specifies the organization, or person in an organization, that assigned and owns the Event id. The responsible_person_organization need not be specified for a particular Event.

4.2.49 Exchange_file

An Exchange_file is a type of File (see 4.2.59) that is either referenced or included in a data exchange. The data associated with an Exchange_file are the following:

— destination_system;

— included_in_exchange.

NOTE Application object EXPRESS:

```
*)
ENTITY exchange_file
     SUBTYPE OF (file);
     destination_system : OPTIONAL system_declaration;
     included_in_exchange : label;
END_ENTITY;
(*
```

4.2.49.1 destination_system

The destination_system specifies the receiving computer system of a data exchange. The destination_system need not be specified for a particular Exchange_file.

4.2.49.2 included_in_exchange

The included_in_exchange specifies if the Exchange_file is included in the data exchange. The value of the included_in_exchange shall be a user-defined text string or one of the following:

— already delivered;

— available for access;

— fully included;

— not deliverable;

— not delivered yet;

— not received;

— partially included;

— receipt accepted;

— receipt acknowledged;

— receipt partially rejected;

— receipt rejected;

— resubmission.

NOTE See 4.2.49.2.1 - 4.2.24.1.12, 4.2.49.2.12 for the definition of each allowable value for included_in_-exchange.

4.2.49.2.1 already delivered
The Tdp_element data that is not in the exchange because it was exchanged previously.

4.2.49.2.2 available for access
The Tdp_element data that is available for the receiving system to remotely access the data.

4.2.49.2.3 fully included
The Tdp_element data and all of data files that are in the exchange.

4.2.49.2.4 not deliverable
The Tdp_element data that are not in the exchange and are not deliverable or planned for delivery.

4.2.49.2.5 not delivered yet
The Tdp_element data that are not in the current exchange and data that will be exchanged in a future exchange

4.2.49.2.6 not received
The Tdp_element data that were not received in a referenced exchange or shipment.

4.2.49.2.7 partially included
Components of the Tdp_element data that are exchanged.

4.2.49.2.8 receipt accepted
The Tdp_element data received in the referenced exchange that were successfully received and accepted as approved by the receiver.

4.2.49.2.9 receipt acknowledgement
The Tdp_element data received in the referenced exchange or shipment that were successfully received

NOTE The receipt_acknowledgement does not imply acceptance by the receiver.

4.2.49.2.10 receipt partially rejected

The Tdp_element data received in the referenced exchange or shipment that were partially rejected due to non-compliance with exchange or delivery requirements.

4.2.49.2.11 receipt rejected

The Tdp_element data received in the referenced exchange or shipment that were rejected due to non-compliance with the exchange or delivery requirements.

4.2.49.2.12 resubmission

The Tdp_element data that are in the exchange as a resubmission or resend

4.2.50 Explicit_graphics

The Explicit_graphics is the geometry that appear on the face of the document. The geometry will be captured in the list presentation description contained within this part of ISO 10303. The data associated with an Explicit_graphics is the following:

— table_geometry.

NOTE Application object EXPRESS:

```
*)
ENTITY explicit_graphics;
  table_geometry : geometry;
END_ENTITY;
(*
```

4.2.50.1 table_geometry

The table_geometry specifies the geometry required to aid in readability of the presentation of lists that are presented in list formats.

4.2.51 External_graphics_file

The External_graphics_file is information concerning the type or data format of the external file, the external file name, and the location for loading the data from the external file.

NOTE 1 External_graphics_file may contain text as well as geometry.

The data associated with an External_graphics_file are the following:

— file_id;

— file_loadpoint_x;

— file_loadpoint_y.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY external_graphics_file;
     file_id : file;
     file_loadpoint_x : a_real;
     file_loadpoint_y : a_real;
END_ENTITY;
(*
```

4.2.51.1 file_id

The file_id specifies the external graphics file and the format.

4.2.51.2 file_loadpoint_x

The file_loadpoint_x specifies the horizontal position of the origin of the External_graphics_file.

EXAMPLE Assuming a page_origin of (0.5, 0.5) for an "A" size sheet, the loadpoint of (7.5, 10.0) would put the vertical position one-half inch from the right side of the page (one-half inch from the lower right corner of the paper).

4.2.51.3 file_loadpoint_y

The file_loadpoint_y specifies the vertical position of the origin of the External_graphics_file.

EXAMPLE Assuming a page_origin of (0.5, 0.5) for an "A" size sheet, the loadpoint of (7.5, 10.0) would put the vertical position one-half inch from the bottom of the page (one-half inch from the lower right corner of the paper).

4.2.52 External_library_reference

The External_library_reference is a mechanism to refer to an entry in an external library other than ISO 13584.

The data associated with an External_library_reference are the following:

— description;

— external_id;

— library_type.

NOTE Application object EXPRESS:

```
*)
ENTITY external_library_reference;
     description : OPTIONAL text;
     external_id : identifier;
     library_type : identifier;
END_ENTITY;
(*
```


4.2.52.1 description

The description specifies the additional information about the External_library_reference. The description need not be specified for a particular External_library_reference..

4.2.52.2 external_id

The external_id specifies the unique identifer of the referenced entry in the external library.

4.2.52.3 library_type

The library_type specifies the kind of library that is being referenced .

4.2.53 Field_content_holder

The Field_content_holder is the data content for the document with the style. Each Field_content_holder is either a Field_content_holder_singular (see 4.2.54) or a Field_content_holder tabulation (see 4.2.55). The data associated with a Field_content_holder are:

— field_justification;

— location;

— text_style.

NOTE Aplication object EXPRESS:

```
*)
ENTITY field_content_holder
     ABSTRACT SUPERTYPE OF
     (ONEOF(field_content_holder_singular,
            field_content_holder_tabulation));
     field_justification : text;
     location : field_location;
     text_style : label;
END_ENTITY;
(*
```

4.2.53.1 field_justification

The field_justification specifies the definition of the justification that indicates the location of the text in the field provided.

EXAMPLE 1 Left justified.

EXAMPLE 2 Centered.

4.2.53.2 text_style

The text_style specifies the formatting information needed for proper instantiation on the associated list. Text_style provides the information necessary for font and printer control information.

EXAMPLE The following information may name a part:

'GASKET (small nozzle body)'
However, the data is required to appear on the parts list as:
GASKET
(small nozzle body)

4.2.53.3 location

The location specifies the location information for the data content.

4.2.54 Field_content_holder_singular

The Field_content_holder_singular is a type of Field_content_holder (see 4.2.53) that is the data content of the header fields. The data associated with a Field_content_holder_singular are the following:

— field_text.

NOTE Application object EXPRESS:

```
*)
ENTITY field_content_holder_singular
  SUBTYPE OF (field_content_holder);
  field_text : text;
END_ENTITY;
(*
```

4.2.54.1 field_text

The field_text specifies Text that is required for instantiation in the header of the associated list. The formatting information will be contained in the style parameter information acquired through the Field_-content_holder.

4.2.55 Field_content_holder_tabulation

The Field_content_holder_tabulation is a type of Field_content_holder (see 4.2.53) that specifies the data content of the table fields (for both header and body data). The data associated with a Field_content_holder tabulation are the following:

— field_text.

NOTE Application object EXPRESS:

```
*)
ENTITY field_content_holder_tabulation
  SUBTYPE OF (field_content_holder);
  field_text : LIST [1:?] OF text;
END_ENTITY;
(*
```

4.2.55.1 field_text

The field_text specifies Text that is required for instantiation on a table (in the header or body) of the associated list. The formatting information is contained in the style parameter information acquired through the Field_content_holder. There may be more than one field_text for a Field_content_holder_tabulation.

4.2.56 Field_location

The Field_location is the location, justification, and boundaries of each of the field titles. Each Field_-location may be one of the following: a String_location_definition (see 4.2.164) or a Field_location_-definition (see 4.2.57).

NOTE Application object EXPRESS:

```
*)
ENTITY field_location
  SUPERTYPE OF
  (ONEOF(string_location_definition,field_location_definition));
END_ENTITY;
(*
```

4.2.57 Field_location_definition

The Field_location_definition is a type of Field_location (see 4.2.56) that specifies the field size in which the text resides and the justification. The data associated with a Field_location_definition are the following:

— field_load_point.

NOTE Application object EXPRESS:

```
*)
ENTITY field_location_definition
  SUBTYPE OF (field_location);
  field_load_point : load_point;
END_ENTITY;
(*
```

4.2.57.1 field_load_point

The field_load_point specifies the load_point that defines the location of the text field.

4.2.58 Field_title

The Field_title is the text and specific location information for the document header, table header, column header, group parent header, and the group parent column header. The data associated with a Field_title are the following:

— field_justification;

— field_text;

— location;

— field_style.

NOTE Application object EXPRESS:

```
*)
ENTITY field_title;
  field_justification : text;
  field_style : label;
  field_text : text;
  location : field_location;
END_ENTITY;
(*
```

4.2.58.1 field_justification

The field_justification specifies the definition of the justification that indicates the location of the text in the field provided.

EXAMPLE 1 Left justified.

EXAMPLE 2 Centered.

4.2.58.2 field_style

The field_style specifies the text formatting information.

EXAMPLE An example of a field_style may be a font.

4.2.58.3 field_text

The field_text specifies the title data information.

4.2.58.4 location

The location specifies the text placement information for the titles or content data.

4.2.59 File

A File is a set of computerized digital data that has meaning based on a predefined digital file format standard. Each File shall be one of the following: a Source_file (see 4.2.153), an Accessed_file (see 4.2.3), or an Exchange_file (see 4.2.49). The data associated with a File are the following:

— assigned_to;

— change_status;

— context_file_format;

— context_file_name;

— distribution_authorizations;

— file_content_property;

— file_content_type;

— file_note;

— native_format_file_name;

— security_identifications;

— size;

— source_system.

NOTE 1 Application object EXPRESS:

```
*)
ENTITY file
  SUPERTYPE OF (ONEOF (accessed_file, exchange_file, source_file));
      assigned_to : OPTIONAL SET [1:?] OF person_and_organization;
      change_status : OPTIONAL change_identification;
      context_file_format : OPTIONAL file_format;
      context_file_name : OPTIONAL identifier;
      distribution_authorizations : OPTIONAL LIST [1:?] OF
        distribution_notice;
      file_content_property : OPTIONAL content_property;
      file_content_type : OPTIONAL label;
      file_note : OPTIONAL notation;
      native_format_file_name : OPTIONAL identifier;
      security_identifications : OPTIONAL LIST [1:?] OF
         security_classification;
      size : OPTIONAL size_characteristics;
      source_system : OPTIONAL system_declaration;
```



```
WHERE
  WR1: EXISTS(context_file_name) OR EXISTS(native_format_file_name)
       OR ('AP232_ARM.ACCESSED_FILE' IN TYPEOF(self));
END_ENTITY;
(*
```

NOTE 2 WHERE Rule states that a file must have a context_file_name, or native_format_file_name, or be an accessed_file.

4.2.59.1 assigned_to

The assigned_to specifies a person and organization that has an association with the File. The assigned_to need not be specified for a particular File. There may be more than one assigned_to for a File.

4.2.59.2 change_status

The change_status specifies the latest alteration made to the File. The change_status need not be specified for a particular File.

4.2.59.3 context_file_format

The context_file_format specifies the computer system file format for the context_file_name. The context_-file_format need not be specified for a particular File.

NOTE The context_file_format is typically either an industry standard file format for data exchange or a mutually agreed to format for data exchange within a business relationship.

EXAMPLE The context_file_format is ISO 10303-21.

4.2.59.4 context_file_name

The context_file_name specifies the computer system file identification for the File.

NOTE When a file is being exchanged, the context_file_name is the file's identification.

4.2.59.5 distribution_authorizations

The distribution_authorizations specifies the industrial or organizational circulation of the information contained in the File. The distribution_authorizations need not be specified for a particular File. There may be more than one distribution_authorizations for a File.

4.2.59.6 file_content_property

The file_content_property specifies characteristics about the the kind of information that is being captured in the File. The file_content_property need not be specified for a particular File.

4.2.59.7 file_content_type

The file_content_type specifies the class of information that being captured in the File. The file_content_type need not be specified for a particular File.

4.2.59.8 file_note

The file_note specifies informational text about the File. The file_note need not be specified for a particular File.

4.2.59.9 native_format_file_name

The native_format_file_name specifies the originating computer system application file identification for the File. The native_format_file_name need not be specified for a particular File.

NOTE The originating computer system file identification is generally referred to as the native file name or the native application file name.

EXAMPLE The native CAX system file name is ‘Fred Drawing for 16PR006'.

4.2.59.10 security_identifications

The security_identifications specifies the business, organizational, or government security or sensitivity of the File and its contents. The security_identifications need not be specified for a particular File. There may be more than one security_identifications for a file.

4.2.59.11 size

The size specifies statistics related to the computer system representation of the File. The size need not be specified for a particular File.

4.2.59.12 source_system

The source_system specifies the identification of the computer system that generated the File. The source_-system need not be specified for a particular File.

4.2.60 File_format

The File_format is the configuration definition of the digital data in the File (see 4.2.59). The format defines the industry or business relationship defined methodology for transcribing the data to the File. The data associated with a File_format are the following:

— change_level;

— format_code;

— format_standard;

— release_date;

— revision_level.

NOTE 1 This format is defined by an industry standard or other agreed to format.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY file_format;
  change_level : OPTIONAL identifier;
  format_code : OPTIONAL identifier;
  format_standard : OPTIONAL text;
  release_date : OPTIONAL date;
  revision_level : OPTIONAL identifier;
WHERE
     WR1: EXISTS (format_code) OR EXISTS (format_standard)
     OR EXISTS (revision_level) OR EXISTS (change_level)
     OR EXISTS (release_date);
END_ENTITY;
(*
```

4.2.60.1 change_level

The change_level specifies a specific alteration made to the format_standard that has been adopted as part of the format standard, but not fully incorporated as part of the format_standard documentation. The change_level need not be specified for a particular File_format.

NOTE For format standards, a change level may be the same as an addendum or amendment.

4.2.60.2 format_code

The format_code specifies a file character encoding. The format_code need not be specified for a particular File_format.

NOTE Industrial practices utilize industry, government, or international standards for data exchange.

EXAMPLE 'ASCII', 'ISO646IRV', 'ISO8859-1', 'EBCDIC', or 'BINARY' are examples of format_code.

4.2.60.3 format_standard

The format_standard specifies the file structure encoding that defines the methodology for transcribing the data to the File. The format_standard need not be specified for a particular File_format.

EXAMPLE 'ISO10303-21', 'JPEG', 'TIFF', 'RTF', or 'HTML' are examples of format_standard.

4.2.60.4 release_date

The release_date specifies the date the version of format_standard was released or issued. The format_-standard need not be specified for a particular File_format.

EXAMPLE Figure 4 illustrates the format style for release_date.

4.2.60.5 revision_level

The revision_level specifies the version of the format_standard from its original release or issue. The revision_level need not be specified for a particular File_format.

NOTE Revisions reflect changes made to the original standard requiring the revision level to advance. For format standards, a revision level may be the same as a version level.

4.2.61 File_relationship

The File_relationship is the association of two Files (see 4.2.59) that supports some business need.

NOTE 1 The semantics of the association is defined by the name and description attributes.

The data associated with a File_relationship are the following:

— description;

— name;

— related_file;

— relating_file.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY file_relationship;
  description : OPTIONAL text;
  name : label;
  related_file : file;
  relating_file : file;
END_ENTITY;
(*
```

4.2.61.1 description

The description specifies the text that describes the File. The description need not be specified for a particular File.

4.2.61.2 name

The name specifies the type of relationship between the Files.

EXAMPLE 'addition', 'copy', 'derivation', 'decomposition', 'peer', 'reference', 'sequence', 'substitution' or 'transition' are examples of name for a File_relationship.

4.2.61.3 related_file

The related_file specifies one of the files in the relationship. If one of the files is subordinate or dependent on the other, the related_file will be the subordinate one.

4.2.61.4 relating_file

The relating_file specifies one of the files in the relationship. If one of the files is subordinate or dependent on the other, the relating_file will be the independent one.

4.2.62 Foreign_defined_item

A Foreign_defined_item is a type of Item_usage (see 4.2.96) that defines the usage of an Item (see 4.2.89) that is used within the context of a drawing and related parts list. The Item that the Foreign_defined_item is describing is defined on a drawing other than the one describing the Foreign_defined_item usage.

NOTE Application object EXPRESS:

```
*)
ENTITY foreign_defined_item
  SUBTYPE OF (item_usage);
END_ENTITY;
(*
```

4.2.63 General_item_definition_relationship

The General_item_definition_relationship is a type of Locally_defined_item (see 4.2.101) that specifies the relationship of one Item to another Item with the meaning of the relationship defined by the attribute 'relation_type'. The data associated with a General_item_definition_relationship are the following:

— base_item_definition;

— description;

— relation_type.

EXAMPLE 1 For a part's definition being replaced by another definition, the following would be an example. For example., for a given part, the design department creates an initial rough definition. This definition is sent to a partner that will work on it and provide a complete definition. This definition becomes the official one and replaces the first one, when it is approved.

EXAMPLE 2 For identifying that one part's definition was drived from another part's definition.

NOTE Application object EXPRESS:

```
*)
ENTITY general_item_definition_relationship
  SUBTYPE OF (locally_defined_item);
  base_item_definition : item;
  description : OPTIONAL TEXT;
  relation_type : label;
END_ENTITY;
(*
```

4.2.63.1 base_item_definition

A base_item_definition specifies the Item that is being replaced or the initial Item that a new Item is being derived from.

4.2.63.2 description

The description specifies additional information about the General_item_definition_relationship. The description need not be specified for a particular General_item_definition_relationship.

4.2.63.3 relation_type

The relation_type specifies the meaning of the relationship. When applicable, the following values shall be used:

— ‘replacement definition’: the related definition replaces the relating definition.

— ‘derivation’: the related definition is derived from the relating definition.

4.2.64 Geometric_validation_property

The Geometric_validation_property is the chaaracteristics of a geometry model that can be used to validate geometric translation results.

The data associated with a Geometric_validation_property are the following:

— property_type;

— property_value.

NOTE Application object EXPRESS:

```
*)
ENTITY geometric_validation_property;
  property_type : label;
  property_value : a_real;
END_ENTITY;
(*
```


4.2.64.1 property_type

The property_type specifies the type of characteristic that is specified for the object

EXAMPLE 1 'centrod' identifies the center point of the mass of a body. The relative position of this point within the body is an invariant datum relative to rotation and translation.

EXAMPLE 2 'surface area' identifies the overall surface of the bodies contained in the referenced shape.

EXAMPLE 3 'volume' identifies the overall volume of the bodies contained in the referenced shape.

4.2.64.2 property_value

The property_value specifies the value that is given for a paricular characteristic.

4.2.65 Geometry

The Geometry is the mathematical representation of shape for a product or annotation.

NOTE Application object EXPRESS:

```
*)
ENTITY geometry;
END_ENTITY;
(*
```

4.2.66 Group_parent_column_header

The Group_parent_column_header is the text information, style, and location of the origin of the header data for a specific group of columns.

The data associated with a Group_parent_column_header are the following:

— single_group_parent_column_header_titles.

NOTE Application object EXPRESS:

```
*)
ENTITY group_parent_column_header;
  single_group_parent_column_header_titles : LIST [1:?] OF field_title;
END_ENTITY;
(*
```

4.2.66.1 single_group_parent_column_header_titles

The single_group_parent_column_header_titles specifies the title data for the group parent columns. There may be more than one single_group_parent_column_header_titles for a Group_parent_column_header.

4.2.67 Group_parent_content_holder

The Group_parent_content_holder is the information needed to place the group parent data content on the document (pointers to the placement location and data to be instantiated). The data associated with a Group_parent_content_holder are the following:

— group_parent_content_presentation.

NOTE Application object EXPRESS:

```
*)
ENTITY group_parent_content_holder;
  group_parent_content_presentation : field_content_holder_tabulation;
END_ENTITY;
(*
```

4.2.67.1 group_parent_content_presentation

The group_parent_content_presentation specifies the data content from the list information and the formatting and placement information necessary for instantation.

4.2.68 Group_parent_header

The Group_parent_header is the text information, style, and location for the titles in the group header data. Group data is a special class of data in which the header data may not be able to be instantiated at once. The data associated with a Group_parent_header are the following:

— single_group_parent_header_titles.

EXAMPLE A parts listing where there are only columns for four parts numbers when ten might be required.

NOTE Application object EXPRESS:

```
*)
ENTITY group_parent_header;
  single_group_parent_header_titles : LIST [1:?] OF field_title;
END_ENTITY;
(*
```

4.2.68.1 single_group_parent_header_titles

The single_group_parent_header_titles specifies the text title data and specifies the formatting and placement information. There may be more than one single_group_parent_header_titles for a Group_-parent_header.

4.2.69 Header

The Header is the configuration management and data management information required for documents and Tdp_elements. The data associated with a Header are the following:

— document_abstract;

— document_keywords;

— header_configuration;

— language;

— sheet_count;

— size_of_sheet.

NOTE Application object EXPRESS:

```
*)
ENTITY header;
  document_abstract : OPTIONAL text;
  document_keywords : OPTIONAL SET [1:?] OF text;
  header_configuration :
       header_configuration_with_element_identification;
  languages: OPTIONAL SET [1:?] OF label;
  sheet_count : OPTIONAL an_integer;
  size_of_sheet : OPTIONAL size_characteristics_full_size;
END_ENTITY;
(*
```

4.2.69.1 document_abstract

The document_abstract specifies a brief overview of the document or Tdp_element. The document_abstract need not be specified for a particular Header.

4.2.69.2 document_keywords

The document_keywords specifies words that describe the main thought or topic of a document or Tdp_-element. The document_keywords need not be specified for a particular Header. There may be more than one keywords for a Header.

4.2.69.3 header_configuration

The header_configuration specifies both the configuration control and data management characteristics required for documents and Tdp_elements.

4.2.69.4 languages

The languages specifies the type of vocabulary nomenclatures used in the document or Tdp_element text. The langauges need not be specified for a particular Header. There may be more than one languages for a Header.

4.2.69.5 sheet_count

The sheet_count specifies the total number of sheets contained within a document or Tdp_element that is page oriented. The sheet_count need not be specified for a particular Header.

NOTE 1 The sheet_count is maintained for status accounting of Associated_lists by sheet and to indicate the general size of the Associated_list.

NOTE 2 Where the presentation form of the data is not defined or exchanged, the sheet_count is optional.

4.2.69.6 size_of_sheet

The size_of_sheet specifies the presentation size of a document page or sheet. The size_of_sheet need not be specified for a particular Header.

NOTE 1 The size_of_sheet is maintained to indicate the general physical size of the Tdp_element's presentation.

NOTE 2 Where the presentation form of the data is not defined or exchanged, the size_of_sheet is optional.

4.2.70 Header_block

The Header_block is the placement location, header text, and style information for the header blocks. The Header_block is instantiated separately for each individual block in the Header (see 4.2.69).

NOTE 1 Only the header data is in block form. The contents are in tables.

The data associated with a Header_block are the following:

— single_block_content_presentation;

— single_block_titles.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY header_block;
     single_block_content_presentation : OPTIONAL LIST [1:?] OF
      field_content_holder_singular;
     single_block_titles : OPTIONAL LIST [1:?] OF field_title;
WHERE
     WR1: EXISTS (single_block_content_presentation) OR EXISTS
     (single_block_titles);
END_ENTITY;
(*
```


4.2.70.1 single_block_content_presentation

The single_block_content_presentation specifies the text, style, and placement information for the list data. There may be more than one single_block_content_presentation for a Header_block.

4.2.70.2 single_block_titles

The single_block_titles specifies the text, style, and placement information for the form or constant information in a header block. The single_block_titles need not be specified for a particular Header_block. There may be more than one single_block_titles for a Header_block.

4.2.71 Header_configuration_with_element_identification

The Header_configuration_with_element_identification is both the configuration control identification and the data management characteristics for an Item (see 4.2.89), document, Tdp_element (see 4.2.170), or File (see 4.2.59). The data associated with a Header_configuration_with_element_identification are the following:

— data_configuration;

— identification.

NOTE Application object EXPRESS:

```
*)
ENTITY header_configuration_with_element_identification;
  data_configuration : OPTIONAL configuration;
  identification : element_identification;
END_ENTITY;
(*
```

4.2.71.1 data_configuration

The data_configuration specifies data management information required to manage and utilize an Item, document, Tdp_element, or File. There may be more than one data_configuration for a header_-configuration_with_element_identification.

4.2.71.2 identification

The identification specifies the configuration control identification of an Item , document, Tdp_element or File.

NOTE For Associated_lists, this identifies the list number, list title, and change activity against the associated list.

4.2.72 Header_presentation

The Header_presentation is the presentation information necessary to format the Header (see 4.2.69) data. The presentation information is content, text, and location. The data associated with a Header_presentation are the following:

— header_blocks_presentation;

— header_tabulation_presentation.

NOTE 1 There may be two sets of header information - one for the first page and one for the remainder of the document.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY header_presentation;
  header_blocks_presentation : LIST [1:?] OF header_block;
  header_tabulation_presentation : OPTIONAL LIST [1:?] OF tabulation;
END_ENTITY;
(*
```

4.2.72.1 header_blocks_presentation

The header_blocks_presentation specifies the text, style, and placement information for the data contained in a given header block. There may be more than one header_blocks_presentation for a Header_-presentation.

4.2.72.2 header_tabulation_presentation

The header_tabulation_presentation specifies information for tables in the header portion of the document. The header_tabulation_presentation need not be specified for a particular Header_presentation. There may be more than one header_tabulation_presentation for a Header_presentation.

4.2.73 Identifier

An Identifier is an alphanumeric string that designates something to be identified. It may not provide natural-language meaning.

NOTE 1 An identifier need not always be unique.

EXAMPLE Revision-A, NOTE 22, or PDQR00145 are identifiers.

NOTE 2 Application object EXPRESS:

```
*)
TYPE identifier = STRING;
END_TYPE;
(*
```


4.2.74 Indentured_data_list

An Indentured_data_list is a type of Document_list (see 4.2.41) that has the contents of the data organized in a hierarchical structure. The hierarchial structure is defined as a “top-down or breakdown” order. An Indentured_data_list is a tabulation of all Drawings (see 4.2.43), Reference_documents (see 4.2.135), Associated_lists (see 4.2.11), and other essential in-house documents required for design disclosure pertaining to the Tdp_element (see 4.2.170) or Item (see 4.2.89) to which Indentured_data_list applies. The data associated with an Indentured_data_list are the following:

— body;

— header.

NOTE Application object EXPRESS:

```
*)
ENTITY indentured_data_list
  SUBTYPE OF (document_list);
  body : indentured_data_list_body;
  header : indentured_data_list_header;
END_ENTITY;
(*
```

4.2.74.1 body

The body specifies the tabulations that are the constituents and subject of the Indentured_data_list.

4.2.74.2 header

The header specifies sufficient configuration control and data management information for a specific Indentured_data_list.

4.2.75 Indentured_data_list_body

The Indentured_data_list_body is the tabulation of information within an Indentured_data_list (see 4.2.74). The tabulated information consists of sufficient identification of related Tdp_element (see 4.2.170) information to meet design disclosure requirements for the Tdp_element that the Indentured_data_list is associated to. The related Tdp_element information is identified within the context of the Tdp_element that the Indentured_data_list is associated to.

The data associated with an Indentured_data_list_body are the following:

— indentured_tabulation;

— notes_list;

— revision_history;

— standardization_documents_list.

NOTE Application object EXPRESS:

```
*)
ENTITY indentured_data_list_body;
  indentured_tabulation : indentured_data_list_tabulation;
  notes_list : OPTIONAL LIST [1:?] OF notation;
  revision_history : OPTIONAL LIST [1:?] OF revision;
  standardization_documents_list : OPTIONAL LIST [1:?] OF
  element_identification;
END_ENTITY;
(*
```

4.2.75.1 indentured_tabulation

The indentured_tabulation specifies the hierarchically defined listing of Tdp_elements. The indentured method indirectly references the document tabulation for the Indentured_data_list.

4.2.75.2 notes_list

The notes_list specifies a listing of notes that are applicable to the Indentured_data_list. The notes may be referenced by the use of a note reference code in any section of the Indentured_data_list. The notes describe any explanatory remarks or notations that may be useful or informative regarding the subject that referenced the note. The notes_list need not be specified for a particular Indentured_data_list_body. There may be more than one notes_list for an Indentured_data_list_body.

4.2.75.3 revision_history

The revision_history specifies a record of revisions of the Indentured_data_list. The revision_history shall be constructed in a manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_history shall not contain an entry for the original release. The revision_history need not be specified for a particular Indentured_data_list_body. There may be more than one revision_history for an Indentured_data_list_body.

4.2.75.4 standardization_documents_list

The standardization_documents_list specifies a tabulation of standardization documents used, referenced, or cited by entries in the Indentured_data_list. The standardization_documents_list need not be specified for a particular Indentured_data_list_body. There may be more than one standardization_documents_list for an Indentured_data_list_body.

4.2.76 Indentured_data_list_entry

The Indentured_data_list_entry is an entry in the Indentured_data_list_tabulation (see 4.2.78).

NOTE 1 The entry is a Tdp_element or an Item (for example, part component) within a Indentured_data_list_-tabulation. The entry will have a relationship to a document that references it and will have associated information characterizing the entry that would comprise the columnar information in an indentured data list tabulation.

The data associated with an Indentured_data_list_entry are the following:

— available_from;

— effective_on;

— entry;

— entry_configuration;

— entry_content_property;

— entry_item_change_level;

— entry_notes;

— indenture_level;

— retrofit;

— special_conditions;

— superseded_entry.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY indentured_data_list_entry;
  available_from: OPTIONAL LIST [1:?] OF source_identification;
  effective_on : OPTIONAL effectivity;
  entry : list_entry_select;
  entry_configuration: OPTIONAL configuration;
  entry_content_property : OPTIONAL content_property;
  entry_item_change_level : OPTIONAL identifier;
  entry_notes : OPTIONAL LIST [1:?] OF notation;
  indenture_level : OPTIONAL label;
  retrofit : OPTIONAL retrofit_usage;
  special_conditions:   OPTIONAL LIST [1:?] OF special_condition;
  superseded_entry: OPTIONAL identification_select;
END_ENTITY;
(*
```

4.2.76.1 available_from

The available_from specifies a Contract_submission, a Reference_document, or Source_location that procurement source information can be obtained from for the Indentured_data_list_entry. The available_-from need not be specified for a particular Indentured_data_list_entry. There may be more than one available_from for an Indentured_data_list_entry.

EXAMPLE When documents available from the International Standards Organization (ISO) are utilized in the preparation of a TDP, the documents are not delivered but are referenced for the receiving organization to acquire through ISO.

NOTE Application object EXPRESS:

```
TYPE source_identification = SELECT
  (contract_submission,
  reference_document,
  source_location);
END_TYPE;
```

4.2.76.2 effective_on

The effective_on specifies the effectivity that governs the use of the Item. The effective_on need not be specified for a particular Indentured_data_list_entry.

4.2.76.3 entry

The entry specifies an Item_parent_to_item_child_relationship, a Tdp_element_parent_ to_item_child_-relationship, a Tdp_element_parent_to_tdp_element_child_relationship, a Tdp_indentured_tdp_element, an Item_parent_to_tdp_element_ child_relationship, or a Tdp_indentured_item that is the subject of the Indentured_data_list_entry.

NOTE 1 An Item_parent_to_item_child_relationship specifies that the row entry (child) is an Item and that the parent entry is an Item.

NOTE 2 An Item_parent_to_tdp_element_child_relationship specifies that the row entry (child) is a Tdp_ element and that its parent entry is an Item.

NOTE 3 A Tdp_element_parent_to_item_child_relationship specifies that the row entry (child) is an Item and that the parent entry is a Tdp_element.

NOTE 4 The Tdp_indentured_item specifies the top Items of an Item-based tree. This enables the listing of multiple part-based document trees within the body of the list.

NOTE 5 A Tdp_element_parent_to_tdp_element_child_relationship specifies that the row entry (child) is a Tdp_-element and that the parent entry is a Tdp_element.

NOTE 6 The Tdp_indentured_tdp_element specifies the top element of a TDP element-based tree. This enables the listing of multiple document trees within the body of the list.

NOTE 7 Application object EXPRESS:

```
TYPE list_entry_select = SELECT
  (item_parent_to_item_child_relationship,
   item_parent_to_tdp_element_child_relationship,
   tdp_element_parent_to_item_child_relationship,
   tdp_element_parent_to_tdp_element_child_relationship,
   tdp_indentured_item,
   tdp_indentured_tdp_element);
END_TYPE;
```


4.2.76.4 entry_configuration

The entry_configuration specifies the data management characteristics about the Tdp_element or Item in the Indentured_data_list_entry. The entry_configuration need not be specified for a particular Indentured_-data_list_entry.

4.2.76.5 entry_content_property

The entry_content_property specifies characteristics about the the kind of information that is being captured in the Tdp_element in the Indentured_data_list_entry. The entry_content_property need not be specified for a particular Indentured_data_list_entry.

4.2.76.6 entry_item_change_level

The entry_item_change_level specifies the revision, change, or issue level that the Indentured_data_list_-entry was last modified. Entry_item_change_level need not be specified for a particular Indentured_data_-list_entry.

NOTE 1 The entry_item_change_level would not be utilized for the original collection of data in a particular Indentured_data_list.

NOTE 2 The entry_item_change_level may be expanded to accommodate a notation specifying that an entry is new or modified for the current version of the Indentured_data_list (for example, a "new this issue" flag).

4.2.76.7 entry_notes

The entry_notes specifies textual information of interest relative to the Indentured_data_list_entry. The textual information is an explanatory remark or notation regarding the Indentured_data_list_entry. The entry_notes need not be specified for a particular Indentured_data_list_entry. There may be more than one entry_notes for an Indentured_data_list_entry.

4.2.76.8 indenture_level

The indenture_level specifies the level of indentured of the Indentured_data_list_entry from the Item or Tdp_element identified as the first object in the Indentured_data_list_tabulation_tabulation_of_entries. The level of indenture indicates a relative position between the Indentured_data_list_entry and the top level item or Tdp_element that is identified in the tabulation_of_entries. The indenture_level need not be specified for a particular Indentured_data_list_entry.

NOTE 1 The level of indenture is formulated based on the document tree or the item tree from the top level Tdp_-element or item identified in the Indentured_data_list_entry.tabulation_of_entries.

NOTE 2 Industrial practices have the Tdp_element or Item as the first level of indenture, while the Tdp_elements or Item that are referenced from the top level Tdp_element or Item are at the second level of indenture. Lower levels of indenture are specified relative to the top level Tdp_element or Item.

4.2.76.9 retrofit

The retrofit specifies the definition and disposition of the Indentured_data_list_entry that is replacing another Indentured_data_list_entry. A retrofit need not be specified for a particular Indentured_data_list_entry.

4.2.76.10 special_conditions

The special_conditions specifies mutually agreed to conditions that apply to the Indentured_data_list_entry. The mutually agreed to conditions relate to codes or designations for Items or Tdp_elements. Special_-conditions need not be specified for a particular Indentured_data_list_entry. There may be more than one special_conditions for an Indentured_data_list_entry.

4.2.76.11 superseded_entry

The superseded_entry specifies the identification of an Item or Tdp_element that was superseded by the current Indentured_data_list_entry. Superseded_entry need not be specified for a particular Indentured_-data_list_entry.

EXAMPLE ISO 10303-203:2001 supersedes ISO 10303-203:1995.

NOTE Application object EXPRESS:

```
TYPE identification_select = SELECT
  (element_identification,
   item_identification);
END_TYPE;
```

4.2.77 Indentured_data_list_header

An Indentured_data_list_header is the configuration control and data management information necessary to manage and control the Indentured_data_list (see 4.2.74). The data associated with an Indentured_data_-list_header are the following:

— common_header;

— procurement_references.

NOTE Application object EXPRESS:

```
*)
ENTITY indentured_data_list_header;
  common_header : header;
  procurement_references: OPTIONAL LIST [1:?] OF identification_select;
END_ENTITY;
(*
```

4.2.77.1 common_header

The common_header specifies the information required for configuration control and administering the use of the data in the Indentured_data_list.

4.2.77.2 procurement_references

The procurement_references specify an Element_identification or an Item_identification that are Tdp_ elements, documents, or Items that were used as justification for development of the Indentured_data_list. The justification is a business purpose that identifies a Tdp_element, document, or Item that was provided by the organization procuring the data contained in the Indentured_data_list . The business purpose permits the association of different Indentured_data_lists in a hierarchial or other business purpose defined method. The procurement_references need not be specified for a particular Indentured_data_list. There may be more than one procurement_reference for an Indentured_data_list_header.

NOTE Application object EXPRESS:

```
TYPE identification_select = SELECT
  (element_identification,
  item_identification);
END_TYPE;
```

4.2.78 Indentured_data_list_tabulation

The Indentured_data_list_tabulation is a listing of valid Tdp_elements (see 4.2.170) within the Indentured_- data_list (see 4.2.74). The listing identifies the valid Tdp_elements in a hierarchial order that is dependent on the business purpose for the Tdp_elements.

The data associated with an Indentured_data_list_tabulation are the following:

— method_of_entry_tabulation;

— tabulation_of_entries.

NOTE Application object EXPRESS:

```
*)
ENTITY indentured_data_list_tabulation;
  method_of_entry_tabulation : indentured_list_method;
  tabulation_of_entries : LIST [1:?] OF indentured_data_list_entry;
END_ENTITY;
(*
```

4.2.78.1 method_of_entry_tabulation

The method_of_entry_tabulation specifies an Indentured_list_by_part, an Indentured_list_by_document, an Indentured_list_by_part_with_document_references_to_parts that specify how the entries in the tabulation_of_entries are referenced by defining the relationship between the tabulation_of_entries.

NOTE 1 The method_of_entry_tabulation identifies how indentured list entries comprising a document tree, part tree, or combination thereof will be structured and listed.

NOTE 2 The method_of_entry_tabulation is presented as a separate entity from Indentured_data_ list_tabulation to provide a path for the Data_definition_exchange to reference a common set of indenture methods.

NOTE 3 An indentured_list_by_part specifies that the primary method of indenture will be by part (part to part) and all subordinate indentures will be from a part perspective (part to document, and document to referenced document).

NOTE 4 An indentured_list_by_document specifies that the only entry in the indentured list will be documents (document to document).

NOTE 5 An indentured_list_by_part_with_document_references_to_parts specifies that the primary method of indenture will be by part (part to part) but all other types of subordinate indentures will be allowable (part to document, document to document, and document to part).

NOTE 6 Application object EXPRESS:

```
*)
TYPE indentured_list_method = SELECT
  (indentured_list_by_document,
   indentured_list_by_part,
   indentured_list_by_part_with_document_references_to_parts);
END_TYPE;
(*
```

4.2.78.2 tabulation_of_entries

The tabulation_of_entries specifies the list of entries using the method defined in the method_of_entry_ tabulation. There may be more than one tabulation_of_entries for an Indentured_data_list_tabulation.

4.2.79 Indentured_list_by_document

The Indentured_list_by_document is an ordered listing of related documents or Tdp_elements (see 4.2.170). The ordered list is an indentured tabulation of Tdp_elements that are defined by a document tree. The ordered list is a document-based document tree where allowed parent-child relationships include only document-to-referenced document relationships. The data associated with an Indentured_list_by_document are the following:

— top_indentured_tdp_elements.

EXAMPLE A listing of a document based document tree:

System or End Item Drawing or Documentation
.Documentation applicable to only the system or end items
.Assembly drawing X
..Documentation applicable to assembly drawing X
..Subassembly drawing Y referenced on assembly drawing X
...Documentation applicable only to subassembly drawing Y
...Detail documentation for parts referenced on drawing Y
.Assembly drawing Z
(Same breakdown as drawing X above.)

NOTE Application object EXPRESS:

```
*)
ENTITY indentured_list_by_document;
  top_indentured_tdp_elements :
         LIST [1:?] OF tdp_indentured_tdp_element;
END_ENTITY;
(*
```

4.2.79.1 top_indentured_tdp_elements

The top_indentured_tdp_element specifies a list of the Tdp_elements that head or start a document-based document tree. Each top_indentured_tdp_element is the listing of multiple document trees within the Indentured_data_list. There may be more than one top_indentured_tdp_element for an Indentured_list_by_-document.

4.2.80 Indentured_list_by_part

The Indentured_list_by_part is an ordered listing of related Items. The ordered list is an indentured tabulation of referenced documents of Tdp_elements (see 4.2.170) that are ordered by the parts that comprise the top assembly Item (see 4.2.89). The sequence or order of the list captures the assembly relationship between parts. The ordered list is a part-based document tree where allowed parent-child relationships include part to referenced document, part to part, and document to referenced document relationships.

EXAMPLE 1 A listing of a part-based document tree:

System or End Item
. Assembly X
.. Documentation applicable to Assembly X
.. Assembly Y which is a sub-assembly of X
... Documentation applicable to Assembly Y
... Detail parts utilized within Assembly Y
.... Documentation applicable to Detail Parts
.... Stock material that is used to fabricate the Detail Parts
..... Documentation applicable to the Stock Material
. Assembly R
.. (Same breakdown as Assembly X above.)

EXAMPLE 2 Top level documents are deliverable or exchangeable product functional requirements; training specifications; engineering instructions for operations, maintenance, repair, inspection; and drawing or design interpretation standard.

NOTE 1 Where top level documents apply to the entire product or system as opposed to a specific version(s) of a product identified by a part number(s), these top level documents may be listed first as a top level indenture entry prior to the listing of the first top level indenture part number or drawing.

The data associated with an Indentured_list_by_part are the following:

— top_indentured_items.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY indentured_list_by_part;
  top_indentured_items : LIST [1:?] OF tdp_indentured_item;
END_ENTITY;
(*
```

4.2.80.1 top_indentured_items

The top_indentured_items specifies a list of the top elements that head or start a parts-based document tree. There may be more than one top_indentured_items for an Indentured_list_by_part.

NOTE The top_indentured_items enables the listing of multiple document trees within an Indentured_data_list.

4.2.81 Indentured_list_by_part_with_document_references_to_parts

The Indentured_list_by_part_with_document_references_to_parts is an ordered listing of related Items (see 4.2.89). The ordered list is defined as relating items in an ordered list of indentured entries in a tabulation of referenced documents that are defined by a parts-based document tree. The sequence or order of listed entries identifies the assembly relationships between parts, but also allows for referenced components from a Tdp_element (see 4.2.170) to be identified including vendor supplied parts and components shown on the Tdp_element to denote assembly constraints but that are fully disclosed on other Tdp_elements (for example, "For Reference Only" drawings). This indenture method requires a Tdp_element to be defined as a single

list entry followed by denoted parts as subsequent list entries that are in turn followed by list entries of the referenced Tdp_elements related to each denoted part.

EXAMPLE A listing of document based document tree internally decomposed by parts:

System or End Item
.Documentation applicable to only the system or end item
. Document or Drawing for Assembly X
.. Assembly X
... Documentation applicable to Assembly X
... Document or Drawing for sub-assembly Y
.... Assembly Y which is a sub-assembly of X
..... Documentation applicable to Assembly Y
..... Document or Drawing of Detail part Z utilized within Assembly Y
..... Detail part Z
..... Documentation applicable to Detail Part Z
..... Stock material that is used to fabricate the Detail Parts
..... Documentation applicable to the Stock Material
... Document or Drawing for Sub-Assembly R
.... Assembly R which is a Sub-Assembly of X
..... (same breakdown as shown above).

NOTE 1 Where top level documents apply to the entire product or system as opposed to a specific version(s) of a product identified by a part number(s), these top level documents may be listed first as a top level indenture entry prior to the listing of the first top level indenture part number or drawing. Examples of top level documents include deliverable or exchangeable product functional requirements; training specifications; engineering instructions for operations, maintenance, repair, inspection; and drawing or design interpretation standard.

The data associated with an Indentured_list_by_part_with_document_references_to_parts are the following:

— top_indentured.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY indentured_list_by_part_with_document_references_to_parts;
  top_indentured: LIST [1:?] OF item_or_element;
END_ENTITY;
(*
```

4.2.81.1 top_indentured

The top_indentured specifies the item or element that is defined to be the top Item or Tdp_element in the ordered list. There may be more than one top_indentured for an Indentured_list_by_part_with_document_-references_to_parts.

NOTE Application object EXPRESS:

```
*)
TYPE item_or_element = SELECT
  (tdp_indentured_item,
```



```
  tdp_indentured_tdp_element);
END_TYPE;
(*
```

4.2.82 Independent_property

The Independent_property is a property that is independent from the actual association of the property to product data. This allows property definitions of a given type to be associated to a single 'general' identification for the particular type of property. The data associated with an Independent_property are the following:

— allowed_units;

— description;

— id;

— property_source.

NOTE Application object EXPRESS:

```
*)
ENTITY independent_property;
  allowed_units : SET [0:?] OF measure;
  description : OPTIONAL text;
  id : identifier;
  property_source : OPTIONAL external_library_reference;
END_ENTITY;
(*
```

4.2.82.1 allowed_units

The allowed_units specifies the unit or set of units that are accepted. The allowed_units need not be specified for a particular Independent_property. There may be more than one allowed_units for an Independent_property.

4.2.82.2 description

The description specifies the text that describes the Independent_property. The description need not be specified for a particular Independent_property.

4.2.82.3 id

The id specifies the identification of the Independent_property.

4.2.82.4 property_source

The property_source specifies the object that defines the Independent_property. The property_source need not be specified for a particular Independent_property.

4.2.83 Independent_property_relationship

The Independent_property_relationship is the associatin between two Independent_property objects (see 4.2.82).

NOTE 1 Independent_property_relationship may be used to indicate that the value of one Independent_property can be derived from the value of another Independent_property.

The data associated with a Independent_property_relationship are the following:

— description;

— related_property;

— relating_property;

— relation_type.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY independent_property_relationship;
  description : OPTIONAL text;
  related_property : independent_property;
  relating_property : independent_property;
  relation_type : label;
END_ENTITY;
(*
```

4.2.83.1 description

The description specifies additional information about the Independent_property_relationship. The description need not be specified for a particular Independent_property_relationship.

4.2.83.2 related_property

The related_property specifies the second of the two Independent_property objects related by the Independent_property_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

4.2.83.3 relating_property

The relating_property specifies the first of the two Independent_property objects related by the Independent_property_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

4.2.83.4 relation_type

The relation_type specifies the meaning of the Independent_property_relationship.

NOTE Where applicable the following values shall be used:
'dependency': The related Independent_property is dependent upon the relating Independent_property,
'hierarchy': The application object defines a hierarchical relationship where the related Independent_property is on a lower level than the relating Independent_property.

4.2.84 Independent_property_usage

The Independent_property_usage is the usage of an independently defined property by an object or concept that requires it.

The data associated with a Independent_property_usage are the following:

— assigned_property;

— assigned_to.

NOTE Application object EXPRESS:

```
*)
ENTITY independent_property_usage;
  assigned_property : independent_property;
  assigned_to : property_assignment;
END_ENTITY;
(*
```

4.2.84.1 assigned_property

The assigned_property specifies the Independent_property that is to be assigned to some object or concept..

4.2.84.2 assigned_to

The assigned_to specifies the object or concept that the Independent_property is to be assigned.

4.2.85 Index_list

An Index_list is a type of Single_document_list (see 4.2.148) that is a tabulation of Data_lists (see 4.2.31) and subordinate Index_lists pertaining to the Item (see 4.2.89) or Tdp_element (see 4.2.170) to which the Index_list applies. The data associated with a Index_list are the following:

— body;

— list_header.

NOTE Application object EXPRESS:

```
*)
ENTITY index_list
  SUBTYPE OF (single_document_list);
  body : index_list_body;
  list_header : index_list_header;
END_ENTITY;
(*
```

4.2.85.1 body

The body specifies the tabulations that are the constituents and subject of the Index_list.

4.2.85.2 list_header

The list_header specifies sufficient configuration control and data management information data for a specific Index_list.

NOTE Data entries such as Company CODE, drawing size, document number, sheet number, revision or issue number, and nomenclature or description may be considered critical or mandatory information.

4.2.86 Index_list_body

The Index_list_body is the tabulation of information within the Index_list (see 4.2.85). The tabulated information consists of sufficient identification of related Tdp_element (see 4.2.170) information to meet design disclosure requirements for the Tdp_element that is associated to the Index_list. The related Tdp_-element information is identified within the context of the Tdp_element or Item (see 4.2.89) that is associated to the Index_list. The data associated with a Index_list_body are the following:

— index_list_entries;

— index_list_notes;

— revision_history.

NOTE Application object EXPRESS:

```
*)
ENTITY index_list_body;
  index_list_entries : LIST [1:?] OF index_list_entry;
  index_list_notes: OPTIONAL LIST [1:?] OF notation;
  revision_history : OPTIONAL LIST [1:?] OF revision;
END_ENTITY;
(*
```

4.2.86.1 index_list_entries

The index_list_entries specify a list of valid Tdp_elements that are contained within the Index_list. There may be more than one index_list_entries for an Index_list_body.

4.2.86.2 index_list_notes

The index_list_notes specifies a listing of notes that are applicable to the Index_list. Some notes are referenced by the use of a note reference code in any section of the Index_list. The notes describe any explanatory remarks or notations that may be useful or informative regarding the subject that referenced the note. The index_list_notes need not be specified for a particular Index_list_body. There may be more than one index_list_notes for an Index_list_body.

4.2.86.3 revision_history

The revision_history specifies a record of revisions of the Index_list. The revision_history shall be constructed in a manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_history shall not contain an entry for the original release. The revision_history need not be specified for a particular Index_list_body. There may be more than one revision_history for an Index_list_body.

4.2.87 Index_list_entry

An Index_list_entry is an entry in the Index_list_body (see 4.2.86). The data associated with an Index_list_-entry are the following:

— entry_item_change_level;

— entry_notes;

— list_entry.

NOTE Application object EXPRESS:

```
*)
ENTITY index_list_entry;
  entry_item_change_level : OPTIONAL identifier;
  entry_notes : OPTIONAL LIST [1:?] OF notation;
  list_entry : header;
END_ENTITY;
(*
```

4.2.87.1 entry_item_change_level

The entry_item_change_level specifies the current change level of the Index_list_entry. The entry_item_-change_level need not be specified for a particular Index_list_entry.

NOTE 1 Industrial practices utilize the entry_item_change_level to notify users of a change to the line item of interest.

NOTE 2 Industrial practices identify the last change of the respective entry.

4.2.87.2 entry_notes

The entry_notes specifies a list of notes that are applicable to the Index_list_entry. The entry_notes need not be specified for a particular Index_list_entry. There may be more than one entry_notes for an Index_-list_entry.

4.2.87.3 list_entry

The list_entry specifies a Header that specifies the allowable Tdp_element of interest.

NOTE Header provides an identification for Tdp_elements such as data list, indentured data list, and index list.

4.2.88 Index_list_header

An Index_list_header is the configuration management and data management information necessary to manage and control the Index_list (see 4.2.85). The data associated with a Index_list_header are the following:

— common_header.

NOTE Application object EXPRESS:

```
*)
ENTITY index_list_header;
  common_header: header;
END_ENTITY;
(*
```

4.2.88.1 common_header

The common_header specifies the information required for configuration control and administering the use of the data in the Index_list.

4.2.89 Item

An Item is any unit or product including parts, assemblies, equipment, accessories, or attachments.

The data associated with an Item are the following:

— alternates;

— conditions_defined_through_constrained_document;

— conditions_defined_through_simple_reference;

— identification;

— item_configuration;

— item_contexts;

— item_weight;

— notes;

— shape;

— shape_aspects;

— special_conditions.

NOTE Application object EXPRESS:

```
*)
ENTITY item;
  alternates : OPTIONAL LIST [1:?] OF alternate_item;
  conditions_defined_through_constrained_document :
      OPTIONAL LIST [1:?] OF reference_document_usage;
  conditions_defined_through_simple_reference :
      OPTIONAL LIST [1:?] OF reference_document;
  identification : item_identification;
  item_configuration : OPTIONAL configuration;
  item_contexts : OPTIONAL LIST [1:?] OF label;
  item_weight : OPTIONAL weight;
  notes : OPTIONAL LIST [1:?] OF notation;
  shape : OPTIONAL shape_select;
  shape_aspects : OPTIONAL SET [1:?] OF shape_select;
  special_conditions: OPTIONAL LIST [1:?] OF special_condition;
END_ENTITY;
(*
```

4.2.89.1 alternates

The alternates specifies a list of substitute Items that meet the form, fit, and function of the design and may be use as an alternative to the original Item. The alternates need not be specified for a particular Item. There may be more than one alternates for an Item.

4.2.89.2 conditions_defined_through_constrained_document

The conditions_defined_through_constrained_document specifies any type of condition of an Item that can be obtained through the use of constraining the use of a document. Constraining parameters are used to aid in defining the condition of the Item. The conditions_defined_through_constrained_document need not be specified for a particular Item. There may be more than one conditions_defined_through_constrained_-document for an Item.

EXAMPLE 1 Final conditions such as final material or process characteristics.

EXAMPLE 2 Finish conditions such as surface finish or external characteristics (for example, paint).

EXAMPLE 3 Design specifications.

4.2.89.3 conditions_defined_through_simple_reference

The conditions_defined_through_simple_reference specifies any type of condition of an Item that can be obtained through a simple reference to document. The conditions_defined_through_simple_reference need not be specified for a particular Item. There may be more than one conditions_defined_through_simple_-reference for an Item.

EXAMPLE 1 Final conditions such as final material or process characteristics.

EXAMPLE 2 Finish conditions such as surface finish or external characteristics (for example, paint).

EXAMPLE 3 Design specifications.

4.2.89.4 identification

The identification specifies the configuration control information required to identify an Item from other Items.

4.2.89.5 item_configuration

The item_configuration specifies configuration control information pertaining to the Item. The item_-configuration need not be specified for a particular Item.

4.2.89.6 item_contexts

The item_context specifies context information pertaining to the Item. The item_context need not be specified for a particular Item. There may be more than one item_contexts for an Item.

NOTE An item_contexts allows view identification of an Item.

EXAMPLE Engineering and Manufacturing are two different item_contexts.

4.2.89.7 item_weight

The item_weight specifies the heaviness of the Item. The item_weight need not be specified for a particular Item.

4.2.89.8 notes

The notes specifies additional human interpretable information that provides clarity and understanding of the information about the Item. The notes need not be specified for a particular Item. There may be more than one notes for an Item.

NOTE Notes may be identified by letters, codes, symbols, or numbers for a particular business purpose.

4.2.89.9 shape

The shape specifies a Shape_model or a File that contains the overall geometric shape of the Item. The shape need not be specified for a particular Item.

NOTE Application object EXPRESS:

```
*)
TYPE shape_select = SELECT
  (file,
   shape_model);
END_TYPE;
(*
```

4.2.89.10 shape_aspects

The shape_aspects specifies a Shape_model or a File that contains a portion of the overall geometric shape of the Item. The shape_aspects need not be specified for a particular Item. There may be more than one shape_asepcts.

NOTE Application object EXPRESS:

```
TYPE shape_select = SELECT
  (file,
   shape_model);
END_TYPE;
```

4.2.89.11 special_conditions

Special_conditions specifies mutually agreed to conditions that apply to the Item. The mutually agreed to conditions relate to codes or designations for the Item. Special_conditions need not be specified for a particular Item. There may be more than one special_conditions for an Item.

4.2.90 Item_identification

An Item_identification is the identification of an Item (see 4.2.89) for purposes of configuration control. The data associated with an Item_identification are the following:

— alternate_identifications;

— change_status;

— classifications;

— design_activity;

— identifying_number;

— item_certification;

— nomenclature_or_name;

— source_information.

NOTE Application object EXPRESS:

```
*)
ENTITY item_identification;
  alternate_identifications :
      OPTIONAL SET [1:?] OF alternate_identification_item_select;
  change_status : OPTIONAL change_identification;
  classifications : LIST [1:?] OF item_type;
  design_activity : design_authority;
  identifying_number : item_identifying_number_select;
  item_certification : OPTIONAL certification;
  nomenclature_or_name : OPTIONAL label;
  source_information : OPTIONAL source_information_type;
END_ENTITY;
(*
```

4.2.90.1 alternate_identifications

The alternate_identifications specify an Alternate_identification_of_item or an Identifier that provides additional ways that an Item may be identified. The alternate_identifications need not be specified for a particular Item_identification. There may be more than one alternate_identifications for an Item_-identification.

EXAMPLE 1 If a company wants to rename a part they received from a supplier use Alternate_element_-identification option.

EXAMPLE 2 If a company wants to alias the id of a part because the character string is too long use Identifier option.

NOTE Application object EXPRESS:

```
*)
TYPE alternate_identification_item_select = SELECT
  (alternate_identification_of_item,
   identifier);
END_TYPE;
(*
```

4.2.90.2 change_status

The change_status specifies the current level of change activity that an Item_identification has undergone. The change_status need not be specified for a particular Item_identification.

NOTE If the change_status is not specified, then the Item is at the original release or issue.

4.2.90.3 classifications

The classifications specifies a set of product types or categories for an Item. There may be more than one classifications for an Item_identification.

4.2.90.4 design_activity

The design_activity specifies the design authority responsible for the Item.

4.2.90.5 identifying_number

The identifying_number specifies an Identifier or a Drawing_suffix_number_combination that is an identifier that defines the identification number for the Item as issued by the design_activity of the component.

NOTE 1 Industrial practices employ a practice of concatenating a dash and an alphanumeric to a drawing number to identify an Item. Each Item receiving a unique alphanumeric for that drawing.

NOTE 2 Application object EXPRESS:

```
TYPE item_identifying_number_select = SELECT
  (identifier,
  drawing_suffix_number_combination);
END_TYPE;
```

4.2.90.6 item_certification

The item_certification specifies that an Item has fulfilled the requirements identified in the Certification. The item_certification need not be specified for a particular Item_identification.

4.2.90.7 nomenclature_or_name

The nomenclature_or_name specifies the name, noun phrase, or abbreviated name of the Item. The nomenclature_or_name need not be specified for a particular Item_identification.

4.2.90.8 source_information

The source_information specifies the method for item procurement. The source_information need not be specified for a particular Item_identification.

4.2.91 Item_list

The item_list is a list of item components for a specific usage. The Item_list specifies the usage context of the list.

EXAMPLE 1 A particular definition of an assembly that lists the components that need to be consumed to define the assembly.

EXAMPLE 2 A particular definition of an application list.

EXAMPLE 3 A particular definition of a make from detail part.

EXAMPLE 4 A particular definition to formulate the parts list for an installation drawing.

EXAMPLE 5 A particular definition of a part that is made from stock material.

EXAMPLE 6 A particular definition of an assembly of component instances.

EXAMPLE 7 A particular definition of a kit of parts.

EXAMPLE 8 A particular definition of a list of net changes on a list of parts.

EXAMPLE 9 A particular definition of a drawing identifying the parts found on itself.

EXAMPLE 10 A particular definition of an option that lists the components that need to be consumed to define the option.

The data associated with an Item_list are the following:

— component_of;

— list_for;

— type_of_component_list.

NOTE Application object EXPRESS:

```
*)
ENTITY item_list;
  component_of : LIST[1:?] OF item_usage;
  list_for : item_list_applies_to_select;
  type_of_component_list : component_list_type_select;
END_ENTITY;
(*
```

4.2.91.1 component_of

The component_of specifies the list of Items that make up the Item_list for the content defined in the type_-of_component_list. There may be more than one component_of for an Item_list.

4.2.91.2 list_for

The list_for specifies an Item_identification or the Item_list_drawing that identifies the applicable Tdp_-element or item.

EXAMPLE An assembly or a drawing.

NOTE Application object EXPRESS:

```
*)
TYPE item_list_applies_to_select = SELECT
  (item_identification,
   item_list_drawing);
END_TYPE;
(*
```

4.2.91.3 type_of_component_list

The type_of_component_list specifies the context for the Item_list. The value of the type_of_component_list shall be either a human_interpretable string of characters or shall be one of the following:

— assembly_defined_on_drawing_component_list;

— installation_drawing_component_list;

— item_not_defined_on_drawing_next_higher_assembly_component_list;

— made_from_component_list_of_one;

— net_change_list;

— synthetic_part_number_component_list.

NOTE 1 See for the definition of each allowable value for type_of_component_list.

NOTE 2 Application object EXPRESS:

```
*)
TYPE component_list_type_enumeration = ENUMERATION OF
  (assembly_defined_on_drawing_component_list,
  installation_drawing_component_list,
  item_not_defined_on_drawing_next_higher_assembly_component_list,
  made_from_component_list_of_one,
  net_change_list,
  synthetic_part_number_component_list);
END_TYPE;
(*
```

EXAMPLE One Item_list is to construct an applications list, one is to identify a make from detail part, one is to formulate the parts list for an installation drawing, and one determines the list of components that need to be consumed to define an assembly on a parts list.

4.2.91.3.1 assembly defined on drawing component list

The Item_list that has one or more components in the list and that if there is only one component in the list, its quantity must be greater than 1. In the context of a Parts_list, these components show up as entries on an assembly column with the column header being the resulting Locally_defined_item assembly.

4.2.91.3.2 installation drawing component list
The components within the Item_list that are all referenced to the drawing. It also specifies that an Item_-list_drawing is utilized so that the drawing number which the list applies to will be captured.

4.2.91.3.3 item not defined on drawing next higher assembly component list
How items (usually top level items on a drawing) defined on a drawing are utilized on the next higher assembly. The next higher assembly being defined on a different drawing.

4.2.91.3.4 made from component list of one
The Item_list that has one component in the list and that component has a quantity of one. In the context of a Parts_list, this component may show up as an entry on an assembly column with the column header being the resulting Make_from_relationship.

4.2.91.3.5 net change list
The Item_list that contains the items that have been added, deleted, or modified. In the context of a Parts_-list, this net change list can provide change visibility from one version of a Parts_list to the next version.

4.2.91.3.6 synthetic part number component list
The item that the list applies to that is an Item_identification which satisfies some process requirement.

NOTE 1 In industrial practices, an Item_identification that is not called out on a drawing is typically called a "synthetic part number".

NOTE 2 Application object EXPRESS:

```
*)
TYPE component_list_type_select = SELECT
  (component_list_type_enumeration,
   text);
END_TYPE;
(*
```

4.2.92 Item_list_drawing

An Item_list_drawing is a drawing or drawing sheet and may identify an item on the drawing for the Item_-list (see 4.2.91).

EXAMPLE The drawing is an installation drawing, where all parts are defined on other drawings but referenced on the installation drawing.

The data associated with an Item_list_drawing are the following:

— assembly_item_list_applies_to;

— drawing_number.

NOTE Application object EXPRESS:

```
*)
ENTITY item_list_drawing;
  assembly_item_list_applies_to : OPTIONAL item_identification;
  drawing_number : element_identification;
END_ENTITY;
(*
```

4.2.92.1 assembly_item_list_applies_to

The assembly_item_list_applies_to specifies the identification of the item applicable to the Item_list. The Item shall appear on the drawing or specific drawing sheet identified in drawing_number. The assembly_-item_list_applies_to need not be specified for a particular Item_list_drawing.

4.2.92.2 drawing_number

The drawing_number specifies the drawing or drawing sheet applicable to the Item_list.

4.2.93 Item_parent_to_item_child_relationship

An Item_parent_to_item_child_relationship is the definition of a relationship between two Items (see 4.2.89). The relationship is a hierarchial relationship that defines the Item to Item assembly relationship. The data associated with an Item_parent_to_item_child_relationship are the following:

— child;

— parent.

NOTE Application object EXPRESS:

```
*)
ENTITY item_parent_to_item_child_relationship;
  child : item_identification;
  parent : item_identification;
END_ENTITY;
(*
```

4.2.93.1 child

The child specifies the identification of the Item of interest.

4.2.93.2 parent

The parent specifies the related parent Item of the child.

4.2.94 Item_parent_to_tdp_element_child_relationship

An Item_parent_to_tdp_element_child_relationship is the definition of a relationship between an Item (see 4.2.89) and a Tdp_element (see 4.2.170). The relationship is a hierarchial relationship that defines the Item to Tdp_element relationship. The data associated with an Item_parent_to_tdp_element_child_relationship are the following.

— child;

— parent.

NOTE Application object EXPRESS:

```
*)
ENTITY item_parent_to_tdp_element_child_relationship;
  child : element_identification;
  parent : item_identification;
END_ENTITY;
(*
```

4.2.94.1 child

The child specifies the identification of the Tdp_element of interest.

4.2.94.2 parent

The parent specifies the related parent Item of the child.

4.2.95 Item_type

The Item_type is a classification code utilized to classify the product data contents of an Item_identification. The data associated with an Item_type are the following:

— code_administrator;

— item_code;

— type_of_coding_scheme.

NOTE 1 Item_types are defined within different industries.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY item_type;
   code_administrator : OPTIONAL company;
     item_code : label;
     type_of_coding_scheme : OPTIONAL text;
END_ENTITY;
(*
```


4.2.95.1 code_administrator

The code_administrator specifies the organization that maintains the item type coding schema. The code_-administrator need not be specified for a particular item_type.

4.2.95.2 item_code

The item_code specifies the classification code for an Item_identification.

4.2.95.3 type_of_coding_scheme

The type_of_coding_scheme specifies industrial standards or a business transaction specific method to classify the Item_identification by the information content of the Item. The standards identify a coding scheme by which types of statements can be referenced. The type_of_coding_scheme need not be specified for a particular item_type.

EXAMPLE ISO 11179-6 defines a coding scheme that an industry may use.

4.2.96 Item_usage

An Item_usage is a particular usage and context of the usage of an Item (see 4.2.89). An Item_usage may be one of the following: a Foreign_defined_item (see 4.2.62) or a Locally_defined_item (see 4.2.101). The data associated with an Item_usage are the following:

— being_defined_for;

— effective_on;

— list_item;

— quantity_used_in_next_higher_assembly;

— retrofit;

— special_conditions.

NOTE 1 In the context of a Parts_list presentation, Item_usage represents a row entry. An Item_usage describes the use of an Item relative to the perspective of a drawing or a part.

NOTE 2 An Item may have many Item_usage(s).

NOTE 3 Each Item_usage for an Item may depict a different effectivity range. Each Item_usage for an Item may reflect its role as a component in an assembly, an Item in an installation, or a stock material in a Make_from_-relationship.

NOTE 4 The quantity of an Item for a particular assembly or effectivity are defined.

NOTE 5 Parameters that relate an Item to the Drawing are defined with Item_usage.

NOTE 6 Application object EXPRESS:

```
*)
ENTITY item_usage
 SUPERTYPE OF (ONEOF(locally_defined_item,foreign_defined_item));
  being_defined_for : item;
  effective_on : OPTIONAL effectivity;
  list_item : OPTIONAL list_item_usage;
  quantity_used_in_next_higher_assembly : OPTIONAL quantity;
  retrofit: OPTIONAL retrofit_usage;
  special_conditions: OPTIONAL LIST [1:?] OF special_condition;
END_ENTITY;
(*
```

4.2.96.1 being_defined_for

The being_defined_for specifies the Item for the applicable Item_usage.

4.2.96.2 effective_on

The effective_on specifies the effectivity that governs the use of the Item in the context of the Item_usage. The effective_on need not be specified for a particular Item_usage.

4.2.96.3 list_item

The list_item specifies drawing and configuration control parameters for use of the Item in the context of the Item_usage. The list_item need not be specified for a particular Item_usage.

NOTE List_item information primarily consists of information utilized in Industrial drawing practices.

EXAMPLE Drawing zone and find number.

4.2.96.4 quantity_used_in_next_higher_assembly

The quantity_used_in_next_higher_assembly specifies a Number, a Weight, Volume, or a human_-interpretable text string that is the quantity for the next highest assembly of an Item in the context of an Item_usage.

NOTE 1 This quantity is typically identified in an applications list.

NOTE 2 Each quantity defines the amount of material or components that is required for a particular usage.

NOTE 3 Number specifies the integer quantity of components that is required for an Assembly_relationship or Make_from_relationship. The number will be one for Make_from_relationship.

NOTE 4 Text specifies a non-number value and may represent a quantity of components, a volume, or a weight. Text may reflect notes such as A/R (as-required).

NOTE 5 Volume specifies the volumetric value of an Item that is used as a component. Real values such as ounces and liters can be denoted.

NOTE 6 Weight specifies how heavy an Item is when used as a component. Real values such as pounds, and grams can be denoted.

NOTE 7 Application object EXPRESS:

```
*)
TYPE quantity_type_select = SELECT
  (text,
  weight,
  volume,
  a_number);
END_TYPE;
(*
```

4.2.96.5 retrofit

The retrofit specifies the definition and disposition of the Item that is replacing another Item. A retrofit need not be specified for a particular Item_usage.

4.2.96.6 special_conditions

The special_conditions specifies a characteristic of the Item_usage that is mutually agreed to between parties for a business purpose. Special_conditions need not be specified for a particular Item_usage. There may be more than one special_conditions for an Item_usage.

4.2.97 Label

A Label is the term that something may be referred to. A Label is a text string that represents the human-interpretable name of something and shall have a natural-language meaning that is computer processable.

NOTE 1 A Label may be a name, and acronym, or any set of text that have a specific meaning among industry experts.

EXAMPLE 1 REV is a label to represent the term "revision".

EXAMPLE 2 NOTE or SECRET is a label.

NOTE 2 Application object EXPRESS:

```
*)
TYPE label = STRING;
END_TYPE;
(*
```


4.2.98 List_item_usage

The List_item_usage is drawing and configuration information about an Item (see 4.2.89) typically used in Industrial drawing practices. The data associated with a List_item_usage are the following:

— approvals;

— drawing_zones;

— entry_item_change_level;

— entry_notes;

— find_number;

— item_type;

— reference_designators;

— security_information.

NOTE Application object EXPRESS:

```
*)
ENTITY list_item_usage;
  approvals : OPTIONAL LIST [1:?] OF approval;
  drawing_zones : OPTIONAL LIST [1:?] OF identifier;
  entry_notes: OPTIONAL LIST [1:?] OF notation;
  entry_item_change_level : OPTIONAL identifier;
  find_number : OPTIONAL identifier;
  item_type : OPTIONAL label;
  reference_designators : OPTIONAL LIST [1:?] OF identifier;
  security_information : OPTIONAL security_classification;
WHERE
  WR1: EXISTS (approvals) OR EXISTS (reference_designators)
  OR EXISTS (entry_notes) OR EXISTS (entry_item_change_level)
  OR EXISTS (drawing_zones) OR EXISTS (item_type)
  OR EXISTS (find_number) OR EXISTS (security_information);
END_ENTITY;
(*
```

4.2.98.1 approvals

The approvals specifies the conformation that usage of the Item is valid based on some criteria. The approvals need not be specified for a particular List_item_usage. There may be more than one approval for a List_item_usage.

4.2.98.2 drawing_zones

The drawing_zones specifies the locations of the Item on a drawing in the context of the Item_usage. The drawing_zones need not be specified for a particular List_item_usage. There may be more than one drawing_zones for a List_item_usage.

4.2.98.3 entry_item_change_level

The entry_item_change_level specifies the current change level of an Item used in a specific context. The entry_item_change_level need not be specified for a particular List_item_usage.

4.2.98.4 entry_notes

The entry_notes specifies a list of notes that are applicable to the List_item_usage context. The entry_notes need not be specified for a particular List_item_usage. There may be more than one entry_notes for an List_item_usage.

4.2.98.5 find_number

The find_number specifies an identifying number that is used in lieu of the number on the field of a drawing for the Item which may be in the context of the Item_usage. The find_number may also identify a unique identifier when multiple uses relationships are created. An instance of when you would be required to use this type of find_number is when quantities vary by effectivity. The find_number need not be specified for a particular List_item_usage.

4.2.98.6 item_type

The item_type specifies a label that identifies the classification of the Item of interest during a particular use of the Item. The item_type need not be specified for a particular List_item_usage.

NOTE 1 The item_type may be utilized to identify if the item is an assembly, sub-assembly, detail part, or composite, casting, forging, machined, or any other type of classification.

NOTE 2 The Item_type may be utilized in conjunction with the Item_identification attribute nomenclature_or_-name to classify an item.

4.2.98.7 reference_designators

The reference_designators specifies a designation on the face of a drawing for an Item in the context of the Item_usage. Reference_designators need not be specified for a particular List_item_usage. There may be more than one references_designators for a List_item_usage.

NOTE Reference_designators are typically used in Industrial practices for electrical and electronic equipment.

4.2.98.8 security_information

The security_information specifies the security level and status of an Item based on its usage. Security_-information need not be specified for a particular List_item_usage.

4.2.99 List_presentation

A List_presentation is the graphical information required to present product data to a human. The graphical information consists of tabulated listings, blocks of data, and single items of data. The data associated with a List_presentation are the following:

— body_page1_format;

— body_pagen_format;

— header_page1_format;

— header_pagen_format;

— page_format;

— unit_of_length.

NOTE Application object EXPRESS:

```
*)
ENTITY list_presentation;
  body_page1_format : body_presentation;
  body_pagen_format : OPTIONAL LIST [1:?] OF body_presentation;
  header_page1_format : header_presentation;
  header_pagen_format : OPTIONAL LIST [1:?] OF header_presentation;
  page_format : page_presentation;
  unit_of_length : measure;
END_ENTITY;
(*
```

4.2.99.1 body_page1_format

The body_page1_format specifies the information required for placement of body data on the first page.

4.2.99.2 body_pagen_format

The body_pagen_format specifies the information required for placement of body data on the remaining pages. The body_pagen_format need not be specified for a particular List_presentation. There may be more than one body_pagen_format for a List_presentation.

4.2.99.3 header_page1_format

The header_page1_format specifies the information required for placement of header data on the first page.

4.2.99.4 header_pagen_format

The header_pagen_format specifies the information required for placement of header data on the remaining pages. The header_pagen_format need not be specified for a particular list_presentation. There may be more than one header_pagen_format for a List_presentation.

4.2.99.5 page_format

The page_format specifies the information required for page placement on a sheet. The sheet size and geometry utilized are also specified.

4.2.99.6 unit_of_length

The unit_of_length specifies the type of units length measurement that are recorded for all List_presentation parameters.

4.2.100 Load_point

The Load_point is the x and y load point location for the text or the field. The data associated with a Load_-point are the following:

— load_point_x;

— load_point_y.

NOTE Application object EXPRESS:

```
*)
ENTITY load_point;
  load_point_x : a_real;
  load_point_y : a_real;
END_ENTITY;
(*
```

4.2.100.1 load_point_x

The load_point_x specifies the numeric x value indicating the horizontal position of the text.

4.2.100.2 load_point_y

The load_point_y specifies the numeric y value indicating the vertical position of the text.

4.2.101 Locally_defined_item

A Locally_defined_item is a type of Item_usage (see 4.2.96) that represents an Item (see 4.2.89) that is used and defined within the context of the Item_list_drawing (see 4.2.92). Each Locally_defined_item may be one of the following: a General_item_definition_relationship (see 4.2.63), Make_from_relationship (see 4.2.105), an Assembly_relationship (see 4.2.10) , or a Made_from_stock_material (see 4.2.104).

NOTE 1 The context of usage of an Item in an Item_usage may be a drawing or a product data set.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY locally_defined_item
  SUPERTYPE OF (ONEOF( make_from_relationship,
    assembly_relationship, made_from_stock_material,
    general_item_definition_relationship))
  SUBTYPE OF (item_usage);
END_ENTITY;
(*
```

4.2.102 Location_instance

A Location_instance is an occurrence of an object that has been defined in a different cartesian coordinate space as a geometric model. The Location_instance is an image of this definition into another Cartesian coordinate space where only the location of this copy has to be specified. Additional uniform scaling, rotation, or mirroring information may be applied to this copy..

NOTE 1 In the case where the units of the cartesian coordinate space of the definition are different from the units to be applied to the Location_instance, unit conversion is required. In case of length unit conversion this has to be considered in addition to the scale attribute.

The data associated with a Location_instance are the following:

— geometry_definition;

— id;

— scale;

— transformation_definition.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY location_instance;
  geometry_definition : geometry_select;
  id : identifier;
  scale : OPTIONAL a_real;
  transformation_definition : transformation;
END_ENTITY;
(*
```

4.2.102.1 geometry_definition

The geometry_definition specifies the geometry to be instantiated.

NOTE Application object EXPRESS:

```
*)
TYPE geometry_select = SELECT
  (geometry,
   file);
END_TYPE;
(*
```

4.2.102.2 id

The id specifies the unique identification of Location_instance.

4.2.102.3 scale

The scale specifies the scaling factor for all cartesian coordinate directions. The scaling factor shall be positive. If the scaling factor is omitted it shall be 1.0. The scale need not be specified for a particular Location_instance.

4.2.102.4 transformation_definition

The transformation_definition specifies the cartesian transformation applied to the instance. All transformation_definitions that can be expressed by an orthonormal 2x2 (for 2D) or 3x3 (for 3D) matrix can be applied (for example, rotation or mirroring operations).

4.2.103 Lot_effectivity

A Lot_effectivity is a type of Effectivity (see 4.2.45) that is the use of an Item (see 4.2.89) in a Product_-configuration (see 4.2.122) where the Item is produced as one of an identifiable group. Lots are used when the Items are produced in batches or when important characteristics might vary between production runs.

NOTE 1 The preceding definition is the paraphrased definition that is defined in ISO 10303-203 for Lot_-effectivity.

The data associated with a Lot_effectivity are the following:

— lot_identification;

— lot_size;

— lot_unit_of_measure.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
  lot_identification : identifier;
  lot_size : a_number;
  lot_unit_of_measure : measure;
END_ENTITY;
(*
```


4.2.103.1 lot_identification

The lot_identification specifies the identification of the group of Items that compose a lot.

NOTE The preceding definition is the paraphrased definition from ISO 10303-203 for lot_number.

4.2.103.2 lot_size

The lot_size specifies the quantity of Items within the lot.

NOTE The preceding definition is the paraphrased definition from ISO 10303-203 for lot_size.

4.2.103.3 lot_unit_of_measure

The lot_unit_of_measure specifies the fixed quantity amount in terms that the lot_size is expressed.

NOTE The preceding definition is the paraphrased definition from ISO 10303-203 for lot_size_unit_of_measure.

4.2.104 Made_from_stock_material

A Made_from_stock_material is a type of Locally_defined_item (see 4.2.101) that is being made from Stock_material (see 4.2.161). A Made_from_stock_material is defined within the context of the Item_usage (see 4.2.96). The data associated with a Made_from_stock_material are the following:

— quantity_of_item_made_from_stock;

— size;

— specified_by;

— stock.

NOTE Application object EXPRESS:

```
*)
ENTITY made_from_stock_material
  SUBTYPE OF (locally_defined_item);
  quantity_of_item_made_from_stock : OPTIONAL a_number;
  size : OPTIONAL stock_size;
  specified_by : OPTIONAL LIST [1:?] OF reference_document_usage;
  stock : OPTIONAL stock_material;
END_ENTITY;
(*
```

4.2.104.1 quantity_of_item_made_from_stock

The quantity_of_item_made_from_stock specifies the number of items that can be made out of one piece of stock material. The quantity_of_item_made_from_stock need not be specified for a particular Made_-from_stock_material.

4.2.104.2 size

The size specifies the usage of the stock material. The size need not be specified for a particular Made_-from_stock_material.

4.2.104.3 specified_by

The specified_by specifies the documentation that identifies the material or process used for the definition of the material stock. There may be more than one specified_by for a made_from_stock_material. The specified_by need not be specified for a particular Made_from_stock_material.

4.2.104.4 stock

The stock specifies the material available in an inventory. The stock need not be specified for a particular Made_from_stock_material.

NOTE Material stock is standard size material that is stocked for usage with Industry.

4.2.105 Make_from_relationship

The Make_from_relationship is a type of Locally_defined_item (see 4.2.101) that is the relationship between the definition of an Item (see 4.2.89) and the definition of an object being manufactured out of that Item. A Make_from_relationship is defined within the context of the Item_usage (see 4.2.96). The data associated with a Make_from_relationship are the following:

— item_that_is_acting_as_stock;

— quantity_of_item_made_from_other_item.

NOTE Application object EXPRESS:

```
*)
ENTITY make_from_relationship
 SUBTYPE OF (locally_defined_item);
 item_that_is_acting_like_stock : item;
 quantity_of_item_made_from_other_item : number_with_units;
END_ENTITY;
(*
```

4.2.105.1 item_that_is_acting_as_stock

An item_that_is_acting_as_stock speficies the Item that is defined in the context of the Item_usage. The item_that_acting_as_stock is used to create the Make_from_relationship.

NOTE The context may be a drawing or a product data set.

4.2.105.2 quantity_of_item_made_from_other_item

The quantity_of_item_made_from_other_item sepcifies the amount of an Item that can be made out of another Item acting as stock material.

NOTE The quantity can be a count of how many Items can be made out of another Item. The quantity can also be a fluid measure of how much of one Item can be made from another Item.

4.2.106 Measure

A Measure is a standard measure units specification.

EXAMPLE inch, gram, minute, volt are different types of measure.

NOTE Application object EXPRESS:

```
*)
TYPE measure = STRING;
END_TYPE;
(*
```

4.2.107 Notation

The notation is textual information of interest. The textual information is human interpretable. The data associated with a Notation are the following:

— note;

— note_parameter;

— note_title;

— reference_code;

— referenced_document_in_note;

— type_of_notation.

NOTE Application object EXPRESS:

```
*)
ENTITY notation;
  note : text;
  note_parameter : OPTIONAL LIST [1:?] OF text;
  note_title: OPTIONAL label;
  reference_code : OPTIONAL identifier;
```



```
  referenced_document_in_note : OPTIONAL SET [1:?] OF
                                element_identification;
  type_of_notation : OPTIONAL SET [1:?] OF label;
END_ENTITY;
(*
```

4.2.107.1 note

The note specifies the explanatory remarks or notations that may be useful or informative to a human.

4.2.107.2 note_parameter

The note_parameter specifies a variable within the body of the note. The note_parameter can be numeric or textual. A note_parameter can be used with a note that requires additional information to complete its semantic meaning. A note_parameter can be another note that is referenced by the base note. There may be more than one note_parameter for a Notation. The note_parameter need not be specified for a particular Notation.

4.2.107.3 note_title

The note_title specifies the heading or title of the note. The note_title need not be specified for a particular Notation.

4.2.107.4 reference_code

The reference_code specifies an identifier that is associated with the note. The identifier is utilized to associate the note with a note listed on the face of the drawing (that an Associated_list (see 4.2.11) pertains), a note listed and explained in another part of the body, or another listing source where detailing of notes are described. The reference_code may then be defined or explained in another part of the body, or other specified listing source where detailing of notes are described. The reference_code need not be specified for a particular Notation.

NOTE The reference_code may be used as a referencing method in lieu of describing a complete explanatory remark or notation directly. This may be done to compress output requirements for presentation purposes or because of potentially repeating explanatory remarks and notations frequently. If the reference_code is used as a referencing method, then another Notation must exist that has the note attribute defined with the identical reference_code.

4.2.107.5 referenced_document_in_note

The referenced_document_in_note specifies a document that is referenced in the text of the note. There may be more than one referenced_document_in_note for a Notation. The referenced_document_in_note need not be specified for a particular Notation.

EXAMPLE The referenced_document_in_note may be a material or process specification that is identified in the text of the note.

4.2.107.6 type_of_notation

The type_of_notation specifies the type of notation. The type of notation is a characteristic of a note that identifies how, what, or where a notation is used. There may be more than one type_of_notation for a Notation. The type_of_notation need not be specified for a particular Notation.

EXAMPLE 1 component: A component specifies that a note is referenced to a component.

EXAMPLE 2 drawing: A drawing specifies that a note is referenced to a drawing.

EXAMPLE 3 drawing effectivity flag note: A drawing effectivity flag note specifies that the note is flagged to reference effectivity notations.

EXAMPLE 4 drawing flag note: A drawing flag note specifies the note as a flagged note indicating that it is located in the design area of a drawing specific to one or more parts. Industrial practices utilize flag notes to flag data on the face of the drawing that are not considered in the note listing (that is, General note) on the face of the drawing.

EXAMPLE 5 drawing general note: A drawing general note specifies that the note is a general note from the face of a drawing.

EXAMPLE 6 remark: A remark specifies that the note is a remark to the reference of interest.

EXAMPLE 7 custom note: A note that is specific to an individual design element instance.

EXAMPLE 8 standard note: A note that can be applied to many design element instances.

EXAMPLE 9 recursive note: A note that refers to other notes or documents.

EXAMPLE 10 usage note: A note that applies to an assembly usage relationship.

4.2.108 Number_with_units

The Number_with_units is a numerical value and a unit of measure standard for the numerical value. The data associated with Number_with_units are the following:

— units_of;

— value_of.

EXAMPLE 1 36 Liters

EXAMPLE 2 45 Meters

NOTE Application object EXPRESS:

```
*)
ENTITY number_with_units;
  units_of : measure;
  value_of : a_real;
END_ENTITY;
(*
```


4.2.108.1 units_of

The units_of specifies the measure standard that is used with the value_of.

4.2.108.2 value_of

The value_of specifies the qualitative characteristic within the context of the designated measure standard.

4.2.109 Other_list

The Other_list is a type of Associated_list (see 4.2.11) that defines list information that is defined for a particular business exchange. The data associated with an Other_list are the following:

— list_body;

— list_header.

NOTE 1 The definition of content in an Other_list is defined between two parties exchanging data and is not given sufficient context within this part of ISO 10303.

EXAMPLE 1 A notes list could be a type of other_list.

EXAMPLE 2 Net list, wire list, application list, or note list.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY other_list
  SUBTYPE OF (associated_list);
  list_body : other_list_body;
  list_header : other_list_header;
END_ENTITY;
(*
```

4.2.109.1 list_body

The list_body specifies the tabulations that are the constituents and subject of the Other_list.

4.2.109.2 list_header

The list_header specifies sufficient configuration control and data management information for a specific Other_list.

4.2.110 Other_list_body

An Other_list_body is the tabulation of information within the Other_list (see 4.2.109). The tabulated information consists of sufficient information to meet design disclosure requirements for the Tdp_element (see 4.2.170) that is associated to the Other_list. The related TDP information is identified within the context

of the Tdp_element that is associated to the Other_list. The data associated with an Other_list_body are the following:

— list_body_tables;

— notes;

— revision_history.

NOTE Application object EXPRESS:

```
*)
ENTITY other_list_body;
  list_body_tables : OPTIONAL LIST [1:?] OF other_list_body_table;
  notes: OPTIONAL LIST [1:?] OF notation;
  revision_history : OPTIONAL LIST [1:?] OF revision;
WHERE
  WR1: EXISTS (list_body_tables) OR EXISTS (notes);
END_ENTITY;
(*
```

4.2.110.1 list_body_tables

The list_body_tables specify the primary tabulations of the Other_list that have been agreed upon for the business exchange. There may be more that one list_body_tables for an Other_list_body. The list_body_-tables need not be specified for a particular other_list_body.

4.2.110.2 notes

The notes specifies a listing of notes that are applicable to the Other_list. Some notes are referenced by the use of a note reference code in any section of the Other_list. The notes describe any explanatory remarks or notations that maybe useful or information regarding the subject that referenced the note. The notes need not be specified for a particular Other_list_body. There may be more than one note for an Other_list_body.

4.2.110.3 revision_history

The revision_history specifies a record of revisions of the Other_list. The revision_history shall be constructed in an manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_history shall not contain an entry for the original release. The revision_history need not be specified for a particular Other_list_body. There may be more than one revision_history for an Other_list_body.

4.2.111 Other_list_body_table

The Other_list_body_table is a business exchange agreed to table for exchanging data for the specific business exchange. The data associated with an Other_list_body_table are the following:

— description_of_tabulation;

— tabulation_column_descriptions;

— tabulation_rows.

NOTE Application object EXPRESS:

```
*)
ENTITY other_list_body_table;
  description_of_tabulation : OPTIONAL label;
  tabulation_column_descriptions : OPTIONAL LIST [1:?] OF label;
  tabulation_rows : LIST [1:?] OF tabulation_entry;
END_ENTITY;
(*
```

4.2.111.1 description_of_tabulation

The description_of_tabulation specifies a computer interpretable definition of the Other_list_body_table. The description_of_tabulation need not be specified for a particular Other_list_body_table.

4.2.111.2 tabulation_column_descriptions

The tabulation_column_descriptions specify the agreed upon business exchange content for the respective tabulation_rows. The relationship to the row is defined by ordinal location in the tabulation_entry tabulation_row. The tabulation_column_descriptions need not be specified for a particular Other_list_-body_table. There may be more than one tabulation_column_descriptions for an Other_list_body_table.

4.2.111.3 tabulation_rows

The tabulation_rows specifies a list of rows in the Other_list_body_table. There may be more than one tabulation_rows for an Other_list_body_table.

4.2.112 Other_list_header

The Other_list_header is the configuration control and data management information necessary to manage and control the Other_list (see 4.2.109). The data associated with an Other_list_header are the following:

— common_header.

NOTE Application object EXPRESS:

```
*)
ENTITY other_list_header;
  common_header : header;
END_ENTITY;
(*
```

4.2.112.1 common_header

The common_header specifies the information required for configuration control and administering the use of the data in the Other_list.

4.2.113 Other_reference_document

The Other_reference_document is a type of Reference_document (see 4.2.135) that specifies references to a critical document that may be used for design disclosure and is not a Standard_document (see 4.2.159) or a Specification_document (see 4.2.157).

NOTE Application object EXPRESS:

```
*)
ENTITY other_reference_document
  SUBTYPE OF (reference_document);
END_ENTITY;
(*
```

4.2.114 Page_parameters

The Page_parameters is the overall location parameters for the page. References may be made to page size standard, a sheet size, or an external file to assist in the page definition.

The data associated with a Page_parameters are the following:

— graphics_declaration;

— page_anchor_x;

— page_anchor_y;

— paper_size;

— page_size_code.

NOTE Application object EXPRESS:

```
*)
ENTITY page_parameters;
  page_size_code : OPTIONAL text;
  paper_size : OPTIONAL text;
  page_anchor_x : a_real;
  page_anchor_y : a_real;
  graphics_declaration : OPTIONAL graphics_parameters;
WHERE
  WR1: EXISTS(page_size_code) OR EXISTS(paper_size);
END_ENTITY;
(*
```

4.2.114.1 graphics_declaration

The graphics_declaration specifies the Explicit_graphics or the External_graphics_file as the source of the geometry to be used on the associated list. The graphics information could be defined explicitly within the context of the part of ISO 10303 or it could be an external file. The graphics_declaration need not be specified for a particular Page_parameters.

EXAMPLE 1 The graphics and the lines that form the border and outline for rows, columns, and headers of associated lists.

EXAMPLE 2 Figure 5 is a graphic_declaration for a tabulated listing.

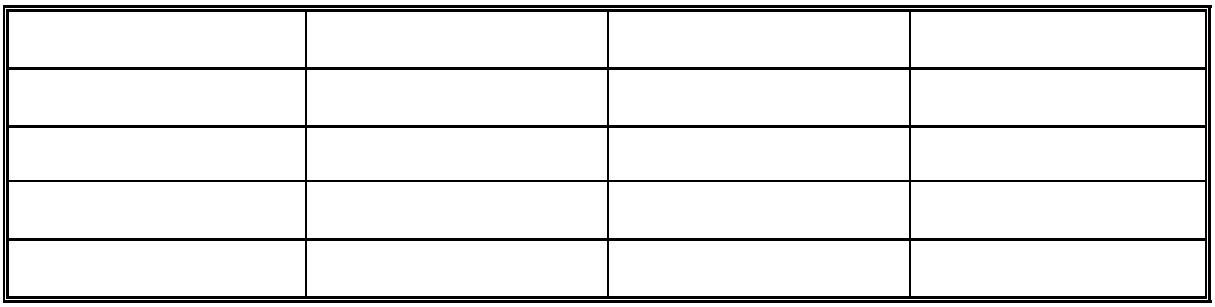

Figure 5 — Graphics utilized in presenting a tabulated listing

NOTE 1 The graphics may or may not be in ISO 10303 format.

NOTE 2 Application object EXPRESS:

```
*)
TYPE graphics_parameters = SELECT
  (explicit_graphics,
   external_graphics_file);
END_TYPE;
(*
```

4.2.114.2 page_anchor_x

The page_anchor_x specifies the horizontal offset from the upper left hand corner of the physical page. The anchor point of (0,0) would be the upper left hand corner of the page (see Figure 6).

NOTE When page_anchor_x is non-zero.

4.2.114.3 page_anchor_y

The page_anchor_y specifies the vertical offset from the upper left had corner of the physical page. The anchor point of (0,0) would be the upper left hand corner of the page (see Figure 6).

4.2.114.4 page_size_code

The page_size_code specifies the page size standard the paper_size is defined within. The page_size_code is used to represent a set of controlling length and width page size parameters. The page_size_code need not be specified for a particular Page_parameters.

EXAMPLE ISO_5457_1980 is a page_size_code.

4.2.114.5 paper_size

The paper_size specifies the size of the length and width of paper to be used in terms of actual linear dimensions. This may be specified when a recognized page standard is not used or when a special sized computer form is used in conjunction with a recognized page size. The paper_size need not be specified for a particular Page_parameters.

4.2.115 Page_presentation

The Page_presentation is the generic page definitions that will set up the location of the data on each page. There is a set of master parameters, specific parameters for the first page, and parameters for the remaining pages. Page1_parameters and pagen_parameters override page_master_parameters. The data associated with a Page_presentation are the following:

— page_master_parameters;

— page1_parameters;

— pagen_parameters.

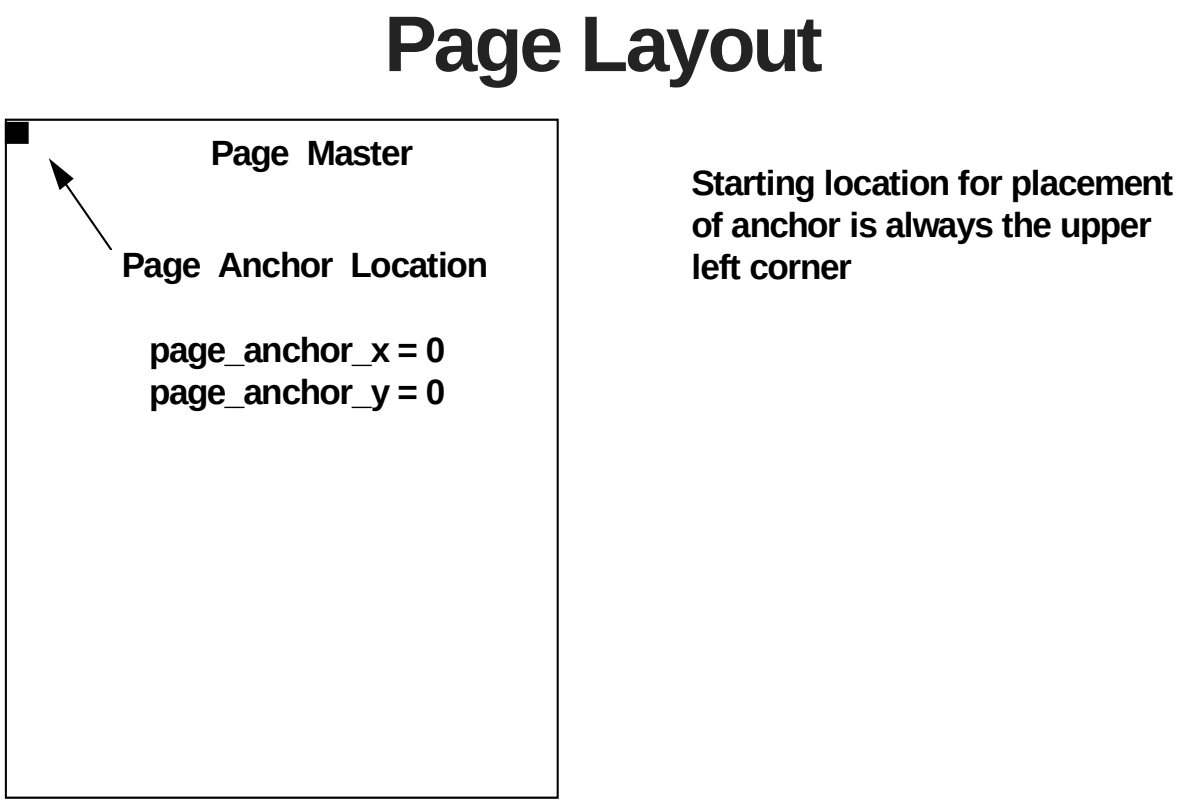

Figure 6 — Page anchor location

NOTE Application object EXPRESS:

```
*)
ENTITY page_presentation;
  page_master_parameters : page_parameters;
  page1_parameters : OPTIONAL page_parameters;
  pagen_parameters : OPTIONAL LIST [1:?] OF page_parameters;
END_ENTITY;
(*
```

4.2.115.1 page_master_parameters

The page_master_parameters specifies the master placement information needed for the entire document.

4.2.115.2 page1_parameters

The page1_parameters specifies the unique placement information needed for the first page. This is optional, as all pages may have the same format information. The page1_parameters need not be specified for a particular Page_presentation.

4.2.115.3 pagen_parameters

The pagen_parameters specifies the placement information needed for page two through the remaining pages of the document. This is optional, as all pages may have the same placement information, or there is only one page. The pagen_parameters need not be specified for a particular Page_presentation. There may be more than one pagen_parameters for a Page_presentation.

4.2.116 Part_occurrence_in_assembly

A Part_occurrence_in_assembly is an Assembly_relationship (see 4.2.10) specifing that one Item (see 4.2.89) uses another Item as a component at a particular location in its assembly. The data associated with a Part_occurrence_in_assembly are the following:

— location.

NOTE Application object EXPRESS:

```
*)
ENTITY part_occurrence_in_assembly
  SUBTYPE OF (assembly_relationship);
  location :OPTIONAL location_instance;
END_ENTITY;
(*
```

4.2.116.1 location

The location specifies the position of the occurrence. The location need not be specified for a particular Part_occurrence_in_assembly.

4.2.117 Parts_list

A Parts_list is a type of Associated_list (see 4.2.11) that is a configuration controlled tabulation of all parts and bulk materials (except those materials that support a process) used in the Item (see 4.2.89). Referenced documents may also be tabulated on parts lists. The data associated with a Parts_list are the following:

— list_body;

— list_header.

NOTE 1 Items listed in subordinate assembly parts lists or specified in a referenced document are not repeated in using the assembly-parts list unless it is necessary to limit operations permitted by the subordinate document. In-house documents, for in-house usage only, may be referenced parenthetically.

NOTE 2 Other designations used to describe a parts list are: list of materials, bill of materials, stock list, and item list.

NOTE 3 Application object EXPRESS:

```
*)
ENTITY parts_list
  SUBTYPE OF (associated_list);
  list_body : parts_list_body;
  list_header : parts_list_header;
END_ENTITY;
(*
```

4.2.117.1 list_body

The list_body specifies the tabulations that are the constituents and subject of the Parts_list.

4.2.117.2 list_header

The list_header specifies sufficient configuration control and data management information for a Parts_list.

4.2.118 Parts_list_body

The Parts_list_body is the tabulations of information within a Parts_list (see 4.2.117). The tabulated information consists of sufficient identification of Items and related information to meet design disclosure requirements for the Item (see 4.2.89) or Tdp_element (see 4.2.170) defined in the Parts_list. The data associated with a Parts_list_body are the following:

— list_notes;

— parts_tabulations;

— referenced_tdp_elements;

— revision_history.

NOTE Application object EXPRESS:

```
*)
ENTITY parts_list_body;
  list_notes : OPTIONAL LIST [1:?] OF notation;
  revision_history : OPTIONAL LIST [1:?] OF revision;
  parts_tabulations : LIST [1:?] OF item_list;
  referenced_tdp_elements :
    OPTIONAL LIST [1:?] OF tdp_element_list_item;
END_ENTITY;
(*
```

4.2.118.1 list_notes

The list_notes specifies a listing of notes that are applicable to the Parts_list. Some notes are referenced by the use of a note reference code in any section of the Parts_list. The notes describe any explanatory remarks or notations that may be useful or informative regarding the subject that referenced the note. Some notes listed are referenced from the Parts_list or from the drawing that the Parts_list pertains. The list_notes need not be specified for a particular Parts_list_body. There may be more than one list_notes for a Parts_list_-body.

NOTE 1 Notes that are applicable to a specific item, component, or assembly are associated to that specific item, component, or assembly.

NOTE 2 The listing of notes is generally done in a sequential order with respect to the codification scheme (or codification schemes) used. A grouping of Parts_list notes and drawing notes should be maintained, along with the order of the listed notes within each referencing scheme.

4.2.118.2 parts_tabulations

The parts_tabulations specify a listing of all parts and bulk materials (except those materials that support a process) used in the item. There may be more than one parts_tabulations for a Parts_list_body.

4.2.118.3 referenced_tdp_elements

The referenced_tdp_elements specifies a listing of the Tdp_elements that are applicable to the Parts_list, the associated drawing, or the associated product data set. The referenced_tdp_elements need not be specified for a particular Parts_list_body. There may be more than one referenced_tdp_elements for a Parts_list_body.

4.2.118.4 revision_history

The revision_history specifies a record of revisions of the Parts_list. The revision_history shall be constructed in a manner whereby the first entry in the tabulation is the first revision, the second entry is the second revision, and subsequent entries in the tabulation are subsequent revisions. The revision_history shall not contain an entry for the original release. The revision_history need not be specified for a particular Parts_list_body. There may be more than one revision_history for a Parts_list_body.

4.2.119 Parts_list_header

A Parts_list_header is the configuration control and data management information necessary to manage and control the Parts_list (see 4.2.117). The data associated with a Parts_list_header are the following:

— common_header.

NOTE Application object EXPRESS:

```
*)
ENTITY parts_list_header;
  common_header : header;
END_ENTITY;
(*
```

4.2.119.1 common_header

The common_header specifies the information required for configuration control and administering the use of the data in the Parts_list.

4.2.120 Person

A Person is an individual with legal rights and duties. A person shall be uniquely identified for purposes of a data exchange. A person may have a first name, a last name, and a title. The data associated with Person are the following:

— first_name;

— last_name;

— middle_name;

— title;

— unique_identifier.

NOTE Application object EXPRESS:

```
*)
ENTITY person;
  first_name : OPTIONAL identifier;
  middle_name : OPTIONAL identifier;
  last_name : OPTIONAL identifier;
  title : OPTIONAL text;
  unique_identifier : identifier;
END_ENTITY;
(*
```


4.2.120.1 first_name

The first_name specifies the first element of the given name of a person. The first_name need not be specified for a particular Person.

4.2.120.2 last_name

The last_name specifies the surname of a person. The last_name need not be specified for a particular Person.

4.2.120.3 middle_name

The middle_name specifies the second given name of a person. The middle_name need not be specified for a particular Person.

4.2.120.4 title

The title specifies the social or professional standing of the person. The title need not be specified for a particular Person.

4.2.120.5 unique_identifier

The unique_identifier specifies a means that may be used to identify a person.

4.2.121 Person_and_organization

A Person_and_organization is an identification of an individual within a Company (see 4.2.17). The data associated with Person_and_organization are the following:

— electronic_mail_address;

— organization;

— person_address;

— person_identification.

NOTE Application object EXPRESS:

```
*)
ENTITY person_and_organization;
  electronic_mail_address : OPTIONAL text;
  organization : OPTIONAL company;
  person_identification : OPTIONAL person;
  person_address : OPTIONAL text;
WHERE
  WR1: EXISTS (organization) OR EXISTS (person_identification);
END_ENTITY;
(*
```

4.2.121.1 electronic_mail_address

The electronic_mail_address specifies the address of a person within an electronic mail system in an organization. The electronic_mail_address need not be specified for a particular Person_and_organization.

4.2.121.2 organization

The organization specifies the organization of a company of interest. The organization need not be specified for a particular Person_and_organization.

4.2.121.3 person_address

The person_address specifies the electronic or physical location where the person at the organizarion can be interfaced with. The person_address need not be specified for a particular Person_and_organization.

4.2.121.4 person_identification

The person_identification identifies the individual of interest within the organization of a Company. The person_identification need not be specified for a particular Person_and_organization.

4.2.122 Product_configuration

A Product_configuration is a variation of a Product_model (see 4.2.127). Configuration management is based on the variation of the Product_model. The data associated with a Product_configuration are the following:

— approval_of_configuration;

— assigned_to;

— configuration_id;

— design_item;

— product_name.

EXAMPLE A Product_configuration for the F14 military aircraft is the D configuration of the F14. The organization has defined four configurations of the F14, A,B,C and D. D, in this instance, is the configuration. A single product model might have many different configurations in order for the organization to identify different variations of it.

NOTE 1 The preceding definition is a different definition from ISO 10303-203 for Product_configuration.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY product_configuration;
  approval_of_configuration : OPTIONAL LIST [1:?] OF approval;
  assigned_to : OPTIONAL SET[1:?] OF person_and_organization;
  configuration_id : identifier;
  design_item : OPTIONAL item_identification;
  product_name : product_model;
END_ENTITY;
(*
```

4.2.122.1 approval_of_configuration

The approval_configuration specifies the list of approvals of the Product_configuration. The approval_of_-configuration need not be specified for a particular Product_configuration. There may be more than one approval_of_configuration for a Product_configuration.

4.2.122.2 assigned_to

The assigned_to specifies a person and organization that has an association with the Product_configuration. The assigned_to need not be specified for a particular Product_configuration. There may be more than one assigned_to for a Product_configuration.

4.2.122.3 configuration_id

The configuration_id specifies the unique identification of a variation of the Product_model.

NOTE The preceding definition is a different definition from ISO 10303-203 for item_id.

4.2.122.4 design_item

The design_item specifies the identification of the item that has been designated as the configuration_item. The design_item need not be specified for a particular Product_configuration.

4.2.122.5 product_name

The product_name specifies the name of the Product_model that is under configuration management.

4.2.123 Product_data_set

The Product_data_set is a type of Tdp_element (see 4.2.170) that is a computer data model that represents a design disclosure representation view of the product Item (see 4.2.89) for the Tdp_element of interest. Each Product_data_set is either a Product_data_set_with_format (see 4.2.124), a Product_data_set_with_-shading (see 4.2.125) or a Product_data_set_without_format (see 4.2.126). The data set defines a characteristic of the product that is of importance to the defining discipline. The data associated with a Product_data_set are the following:

— file_configuration;

— related.

EXAMPLE Geometric shape or associated information.

NOTE Application object EXPRESS:

```
*)
ENTITY product_data_set
  ABSTRACT SUPERTYPE OF
   (ONEOF(product_data_set_with_format,product_data_set_without_format,
      product_data_set_with_shading))SUBTYPE OF (tdp_element);
  related : OPTIONAL LIST [1:?] OF drawing_or_product_data_set;
  file_configuration : OPTIONAL LIST [1:?] OF source_file;
END_ENTITY;
(*
```

4.2.123.1 file_configuration

The file_configuration specifies the list of files that composes the Product_data_set. The file_configuration need not be specified for a particular Product_data_set. There may be more than one file_configuration for a Product_data_set.

4.2.123.2 related

The related specifies a Drawing or a Product_data_set that this model is associated to. This represents information that defines the relationship for design disclosure. The related need not be specified for a particular Product_data_set. There may be more than one related for a Product_data_set.

NOTE Application object EXPRESS:

```
TYPE drawing_or_product_data_set = SELECT
  (drawing,
  product_data_set);
END_TYPE;
```


4.2.124 Product_data_set_with_format

A Product_data_set_with_format is a type of Product_data_set (see 4.2.123) that is a computer interpretable data model with information defined for a human-viewable presentation. The data associated with a Product_data_set_with_format are the following:

— presentation_of_product_data_set;

— product_header.

NOTE Application object EXPRESS:

```
*)
ENTITY product_data_set_with_format
  SUBTYPE OF (product_data_set);
  presentation_of_product_data_set : OPTIONAL LIST [1:?] OF
     product_presentation;
  product_header : header;
END_ENTITY;
(*
```

4.2.124.1 presentation_of_product_data_set

The presentation_of_product_data_set specifies the data set and the presentation of this data set. The presentation_of_product_data_set need not be specified for a particular Product_data_set_with_format.

4.2.124.2 product_header

The product_header specifies the information required for configuration control and administering the use of the data in the Product_data_set_with_format.

NOTE product_header may include sheet or page count information.

4.2.125 Product_data_set_with_shading

The Product_data_set_with_shading is a type of Product_data_set (see 4.2.123) that is a computer model with shading information for the presentation of the product data. The data associated with a Product_data_-set_with_shading are the following:

— a_shaded_model;

— data_configuration.

NOTE Application object EXPRESS:

```
*)
ENTITY product_data_set_with_shading
  SUBTYPE OF (product_data_set);
  a_shaded_model : shaded_shape_model;
```

```
    data_configuration : header_configuration_with_element_identification;
END_ENTITY;
(*
```

4.2.125.1 a_shaded_model

A_shaded_model specifies the shape information in a human viewable format that allows shading of the geometric information that is the Product_data_set.

4.2.125.2 data_configuration

The data_configuration specifies the information required for configuration control and administering the use of the data in the Product_data_set_with_shading.

NOTE Data_configuration does not include sheet or page count information.

4.2.126 Product_data_set_without_format

A Product_data_set_without_format is a type of Product_data_set (see 4.2.123) that is a computer data model without presentation information. The data associated with a Product_data_set_without_format are the following:

— a_model;

— data_configuration.

NOTE Application object EXPRESS:

```
*)
ENTITY product_data_set_without_format
  SUBTYPE OF (product_data_set);
  a_model : OPTIONAL shape_model;
  data_configuration : header_configuration_with_element_identification;
END_ENTITY;
(*
```

4.2.126.1 a_model

The a_model specifies the shape information that is the Product_data_set. The a_model need not be specified for a particular Product_data_set_without_format.

4.2.126.2 data_configuration

The data_configuration specifies the information required for configuration control and administering the use of the data in the Product_data_set_without_format.

NOTE Data_cofiguration does not include sheet or page count information.

4.2.127 Product_model

A Product_model is the product that the organization provides to its customers. The Product_model is identified for planning purposes in the design stage of a product. The data associated with a Product_model are the following:

— model_name;

— projects.

NOTE 1 The preceding definition is the definition from ISO 10303-203 for Product_model.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY product_model;
  model_name : label;
  projects : OPTIONAL SET[1:?] OF project;
END_ENTITY;
(*
```

4.2.127.1 model_name

The model_name specifies the unique identification assigned by an organization to a product that the organization provides to its customers.

NOTE The preceding definition is the definition from ISO 10303-203 for Product_model.

4.2.127.2 projects

The projects specifies the set of projects related to the Product_model. The projects need not be specified for a particular Product_model. There may be more than one projects for a Product_model.

4.2.128 Product_presentation

The Product_presentation is the shape description of the Product with descriptive annotation for human interpretation. The data associated with a Product_presentation are the following:

— annotation_presentation;

— part_shape_views.

NOTE Application object EXPRESS:

```
*)
ENTITY product_presentation;
  annotation_presentation : annotation;
  part_shape_views : OPTIONAL LIST [1:?] OF shape_model;
END_ENTITY;
(*
```


4.2.128.1 annotation_presentation

The annotation_presentation specifies text and non-shape geometry. It can be used in conjunction with product shape presentation.

4.2.128.2 part_shape_views

The part_shape_views specify the viewing parameters for product shape presentation. The part_shape_views need not be specified for a particular Product_presentation. There may be more than one part_shape_views for a Product_presentation.

4.2.129 Project

A Project is a work effort that one or more people and organizations participate in. The data associated with a Project are the following:

— description;

— end_date;

— id;

— name;

— participants;

— project_owner;

— start_date.

NOTE Application object EXPRESS:

```
*)
ENTITY project;
  description : OPTIONAL text;
  end_date : OPTIONAL date;
  id : identifier;
  name : label;
  participants : SET [1:?] OF person_and_organization;
  project_owner : OPTIONAL person_and_organization;
  start_date : OPTIONAL date;
END_ENTITY;
(*
```

4.2.129.1 description

The description specifies the text that describes the project. The description need not be specified for a particular Project.

4.2.129.2 end_date

The end_date specifies the date that the project stops. The end_date need not be specified for a particular Project.

4.2.129.3 id

The id specifies the identification of the project assigned by the organization.

4.2.129.4 name

The name specifies the words by which the project is known.

4.2.129.5 participants

The participants specifies the organization or person and organization that participate in the project. There may be more than one participants for a Project.

4.2.129.6 project_owner

The project_owner specifies the organization or person and organization that assigned and owns the project id. The project_owner need not be specified for a particular Project.

4.2.129.7 start_date

The start_date specifies the date the project begins. The start_date need not be specified for a particular Project.

4.2.130 Promissory_usage

A Promissory_usage is an Assembly_relationship (see 4.2.10) that specifies there is an intention that on Item (see 4.2.89) will use another Item as a component in its assembly.

NOTE Application object EXPRESS:

```
*)
ENTITY promissory_usage
  SUBTYPE OF (assembly_relationship);
END_ENTITY;
(*
```

4.2.131 Property_assignment

A Property_assignment is a property that is assigned to an element of product date. The data associated with a Property_assignment are the following:

— described_element;

— description;

— name.

NOTE Application object EXPRESS:

```
*)
ENTITY property_assignment;
  described_element : identifier;
  description : OPTIONAL text;
  name : OPTIONAL label;
END_ENTITY;
(*
```

4.2.131.1 described_element

The described_element specifies the element of product data that the property is assigned.

4.2.131.2 description

The description specifies the text that characterizes the property. The description need not be specified for a particular Property_assignment.

4.2.131.3 name

The name specifies the words by which the property is known. The name need not be specified for a particular Property_assignment.

4.2.132 Quantified_part_usage_in_assembly

A Quantified_part_usage_in_assembly is an Assembly_relationship (see 4.2.10) specifying that one Item (see 4.2.89) is the direct parent of another in an Item.

NOTE 1 When a part is a direct parent of another part, the parent is the resulting assembly.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY quantified_part_usage_in_assembly
  SUBTYPE OF (assembly_relationship);
END_ENTITY;
(*
```


4.2.133 Quantity

A Quantity is the value of an amount with or without an accuracy parameter. The data associated with a Quantity are the following:

— quantity_accuracy;

— value_of.

NOTE Application object EXPRESS:

```
*)
ENTITY quantity;
  quantity_accuracy : OPTIONAL quantity_accuracy_select;
  value_of : quantity_type_select;
END_ENTITY;
(*
```

4.2.133.1 quantity_accuracy

The quantity_ accuracy specifies the exactness of the value of the quantity. The value of the quantity_-accuracy shall be either a human_interpretable string of characters or shall be one of the following:

— approximate;

— as required;

— exact.

The quantity_accuracy need not be specified for a particular Quantity.

NOTE 1 See 4.2.133.1.1 - 4.2.133.1.3 for the definition of each allowable value for base.

EXAMPLE Exact, as required, approximate.

NOTE 2 Application object EXPRESS:

```
*)
TYPE quantity_accuracy_select = SELECT
  (quantity_accuracy_enumeration,
  text);
END_TYPE;

TYPE quantity_accuracy_enumeration = ENUMERATION OF
  (approximate,
  as_required,
  exact);
END_TYPE;
(*
```

4.2.133.1.1 approximate
Qualifies that the value is a rough idea of how much might be required, but leaves it to the manufacturing process to apply as much as needed with the stipulation that a nonzero quantity of material must be applied. Approximate may be used for bulk material such as sealers and lubricants.

4.2.133.1.2 as required
Qualifies that the manufacturing process can determine the exact quantity and that a zero is a valid quantity.

4.2.133.1.3 exact
Qualifes that the value of the engineering specification of the Quantity is precise.

4.2.133.2 value_of

The value_of specifies A_number, a Weight, Volume, or a human_interpretable text string that is the quantity for the next highest assembly of an Item in the context of an Item_usage.

NOTE 1 This quantity is typically identified in an applications list.

NOTE 2 Each quantity defines the amount of material or components that is required for a particular usage.

NOTE 3 A_number specifies the integer quantity of components that is required for an Assembly_relationship or Make_from_relationship. The number will be one for Make_from_relationship.

NOTE 4 Text specifies a non-number value and may represent a quantity of components, a volume, or a weight. Text may reflect notes such as A/R (as-required).

NOTE 5 Volume specifies the volumetric value of an Item that is used as a component. Real values such as ounces and liters can be denoted.

NOTE 6 Weight specifies how heavy an Item is when used as a component. Real values such as pounds, and grams can be denoted.

NOTE 7 Application object EXPRESS:

```
TYPE quantity_type_select = SELECT
  (text,
  weight,
  volume,
  a_number);
END_TYPE;
```

4.2.134 Reason

A Reason is the rationale for a data exchange and the applicable item that the product information describes. The data associated with a Reason are the following:

— applicable_to;

— base;

— exchange_purpose.

NOTE Application object EXPRESS:

```
*)
ENTITY reason;
  applicable_to : OPTIONAL LIST [1:?] OF item_identification;
  base : OPTIONAL exchange_reason;
  exchange_purpose : OPTIONAL text;
WHERE
  WR1: EXISTS (exchange_purpose) OR EXISTS (base);
END_ENTITY;
(*
```

4.2.134.1 applicable_to

The applicable_to specifies the Item that is pertinent to the exchange. The applicable_to need not be specified for a particular Reason. There may be more than one applicable_to for a Reason.

4.2.134.2 base

The base specifies a collection of reasons for a data exchange. The value of the base shall be one of the following:

— acknowledge receipt of delivery;

— engineering design analysis;

— interim engineering design review;

— final design review;

— initial data submittal;

— procurement design package;

— provisioning data submittal;

— request for proposal;

— request for quote;

— revision update data submittal;

— specification design package.

The base need not be specified for a particular Reason.

NOTE 1 See 4.2.134.2.1 - 4.2.134.2.11 for the definition of each allowable value for base.

EXAMPLE Request for proposal, request for quote, data submittal.

NOTE 2 Application object EXPRESS:

```
*)
TYPE exchange_reason = ENUMERATION OF
  (acknowledge_receipt_of_delivery,
  engineering_design_analysis,
  interim_engineering_design_review,
  final_design_review,
  initial_data_submittal,
  procurement_design_package,
  provisioning_data_submittal,
  request_for_proposal,
  request_for_quote,
  revision_update_data_submittal,
  specification_design_package);
END_TYPE;
(*
```

4.2.134.2.1 acknowledge receipt of delivery

The exchange that is a correspondence from the receiver to the original sender acknowledging the receipt of the data provided by the sender.

4.2.134.2.2 engineering design analysis

The exchange that is a general purpose exchange of design information for general analysis and technical interchange. The exchange does not fulfill contractual obligations and is considered not formal.

EXAMPLE An exchange for concurrent engineering design analysis or comparisons is not a formal contractual obligation.

4.2.134.2.3 final design review

The exchange that is a formal exchange and is intended to fulfill contractual or other obligations. The receiver of the package is expected to conduct a formal review of the package and assess the final design, methods, and standards used in design. The receiver then provides an acceptance or rejection of the design in terms of technical engineering content (not acceptance or rejection of the design documentation as a deliverable).

4.2.134.2.4 initial data submittal

The exchange that is submitted in partial fulfillment of contractual obligations requiring the delivery of design data. The initial submittal is the first formal delivery of configuration managed design data that comprises the design "baseline". The receiver of the package is expected to conduct a formal review of the package and assess the final design, methods, and standards used in design to ensure it meets specified requirements (for example, the support of redesign, repair, remanufacture, or reprocurement). The receiver then provides a formal disposition of acceptance or rejection of the design in terms of both the technical engineering content and design documentation.

4.2.134.2.5 interim engineering design review

The exchange that is a formal exchange intended to fulfill contractual or other obligations. The receiver of the package is expected to assess the preliminary design concepts, methods, and standards and provide general feedback to the sender.

4.2.134.2.6 procurement design package

The exchange that contains the formally released or managed technical design data required to perform a part procurement activity. The package includes a complete design package from which the receiver can manufacture or provide an item.

4.2.134.2.7 provisioning data submittal

The exchange that is submitted in partial fulfillment of contractual obligations requiring the sender to provide design data (at negotiated levels of design documentation completeness) which enables provisioning analysis on the part.

4.2.134.2.8 request for proposal

The exchange that is provided to support the development of a requested proposal involving the designed item.

4.2.134.2.9 request for quote

The exchange that is provided to support the development of an estimate involving the designed item.

4.2.134.2.10 revision update data submittal

The exchange that is submitted in partial fulfillment of contractual obligations requiring the delivery of design data. The revision update data submittal is a formal delivery of configuration managed design data (subsequent to the initial data submittal) that revises the design "baseline". The receiver of the package is expected to conduct a formal review of the package and assess the final design, methods, and standards used in the design to ensure it meets specified requirements (for example, the support of redesign, repair, remanufacture, or reprocurement). The receiver then provides a formal disposition of acceptance or rejection of the design in terms of both the technical engineering content and design documentation.

4.2.134.2.11 specification design package

The exchange that contains the formally released-managed technical design data required to perform a design procurement activity. The package includes a complete set of design specifications (for example, specification control drawings, associated lists, material and process specifications) that the receiver can design the desired item.

4.2.134.3 exchange_purpose

The exchange_purpose specifies textual information where the reason for the exchange can be specified. The exchange_purpose need not be specified for a particular Reason.

4.2.135 Reference_document

A Reference_document is a type of Tdp_element (see 4.2.170) that is a specification, a standard, a directive, a guideline, or some other type of document that is required for full design disclosure reference. Each Reference_document is either a Specification_document (see 4.2.157), a Standard_document (see 4.2.159), or an Other_reference_document (see 4.2.113). The data associated with a Reference_document are the following:

— document_identification.

NOTE Application object EXPRESS:

```
*)
ENTITY reference_document
  ABSTRACT SUPERTYPE OF (ONEOF(specification_document,
   standard_document,other_reference_document))
  SUBTYPE OF (tdp_element);
  document_identification : element_identification;
END_ENTITY;
(*
```

4.2.135.1 document_identification

The document_identification specifies the unique identification of the Reference_document.

4.2.136 Reference_document_usage

A Reference_document_usage is an identification of a constraint that is to be applied with the use of a Reference_document (see 4.2.135). The data associated with a Reference_document_usage are the following:

— document_referenced;

— location_in_document;

— usage_description;

— usage_code;

— usage_parameter.

NOTE Application object EXPRESS:

```
*)
ENTITY reference_document_usage;
  document_referenced : reference_document;
  location_in_document: OPTIONAL label;
  usage_code : OPTIONAL identifier;
  usage_description : OPTIONAL text;
  usage_parameter : OPTIONAL document_usage_parameter;
END_ENTITY;
(*
```

4.2.136.1 document_referenced

The document_referenced specifies the Reference_document that is being constrained.

4.2.136.2 location_in_document

The location_in_document specifies a section, chapter, paragraph, or other delineation within the Reference_document that is of interest. The location_in_document need not be specified for a particular Reference_document_usage.

4.2.136.3 usage_code

The usage_code specifies a unique code within the Reference_document that pertains to the usage of the Reference_document. The usage_code need not be specified for a particular Reference_document_usage.

EXAMPLE The usage_code subjects could be 'class', 'composition', 'form', 'grade', 'type', 'color','density', or 'condition'.

4.2.136.4 usage_description

The usage_description specifies a textual description of the use of the Reference_document within the context of the Reference_document_usage. The usage_description need not be specified for a particular Reference_document_usage.

4.2.136.5 usage_parameter

The usage_parameter specifies the identification and description of a specific piece of information that is contained in a document. The usage_parameter need not be specified for a particular Reference_document_-usage.

4.2.137 Relative_event

The Relative_event is an Event (see 4.2.48) that includes an time offset and an association to a base event. The data associated with the Relative_event are the following:

— offset;

— related_event.

NOTE Application object EXPRESS:

```
*)
ENTITY relative_event;
  offset : number_with_units;
  related_event : event;
END_ENTITY;
(*
```

4.2.137.1 offset

The offset specifies the delta in time between the two events.

4.2.137.2 related_event

The related_event specifies the Event that this Event is base on.

4.2.138 Release_authentication

The Release_authentication is the authentication of the release of the Tdp_element (see 4.2.170) from the originating system. The data associated with the Release_authentication are the following:

— authentication;

— person_responsible;

— release_authority;

— release_authority_code;

— release_date.

NOTE Application object EXPRESS:

```
*)
ENTITY release_authentication;
  release_date : date;
  authentication : identifier;
  person_responsible : OPTIONAL person;
  release_authority : company;
  release_authority_code : OPTIONAL company_code;
END_ENTITY;
(*
```

4.2.138.1 authentication

The authentication specifies the originating system identification of the authentication.

NOTE The authentication constitutes an electronic signature that assures the integrity of the Tdp_element's contents. This may be conducted by reviewing the Tdp_element's contents or by validating the automated data processing system procedures that constructed or manages the Tdp_element.

4.2.138.2 person_responsible

The person_responsible specifies the responsible person that has authenticated the Tdp_element contents for the purposes of release. The person_responsible need not be specified for a particular Release_-authentication.

NOTE In cases where a symbol is used instead of a signature, the authentication should define the symbol name. The graphical presentation form of the symbol is addressed under presentation entities.

4.2.138.3 release_authority

The release_authority specifies the name of the Company responsible for the release of the Tdp_element.

4.2.138.4 release_authority_code

The release_authority_code specifies the Company_code responsible for release of the Tdp_element. The release_authority_code need not be specified for a particular Release_authentication.

4.2.138.5 release_date

The release_date specifies the date and time that the design release activity released the Tdp_element.

EXAMPLE Figure 4 illustrates the format style for release_date.

4.2.139 Retrofit_usage

The retrofit_usage is the definition and disposition of an Item (see 4.2.89) that is replacing another Item that is a component of a product. The replacing Item may be the original Item that has been modified. The data associated with the Retrofit_usage are the following:

— defining_document;

— disposition_for;

— retrofit_description.

NOTE 1 Retrofit items typically change the form, fit, or function of the original item.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY retrofit_usage;
  retrofit_description : OPTIONAL notation;
  defining_document : OPTIONAL other_reference_document;
  disposition_for : retrofit_state;
END_ENTITY;
(*
```

4.2.139.1 retrofit_description

The retrofit_description specifies a characterization of the retrofit. The retrofit_description need not be specified for a particular Retrofit_usage.

4.2.139.2 defining_document

The defining_document specifies a document that defines the actions required for the retrofit. The defining_-document need not be specified for a particular Retrofit_usage.

4.2.139.3 disposition_for

The disposition_for specifies the action to be taken with the original Item. The value of the disposition_for shall be one of the following:

— add;

— delete;

— modify.

NOTE 1 See 4.2.139.3.1 - 4.2.139.3.3 for the definition of each allowable value for disposition_of.

NOTE 2 Application object EXPRESS:

```
*)
TYPE retrofit_state = ENUMERATION OF
  (add,
   delete,
   modify);
END_TYPE;
(*
```

4.2.139.3.1 add
add specifies that the Item was added as a result of the retrofit.

4.2.139.3.2 delete
delete specifies that the Item was deleted as a result of the retrofit.

4.2.139.3.3 modify
modify specifies that the Item was modified as a result of the retrofit.

4.2.140 Revision

The Revision is the current revision information of the Item (see 4.2.89) or document of interest. The data associated with the Revision are the following:

— authorizing_documents;

— revision_approval;

— revision_date;

— revision_description;

— revision_level.

NOTE Application object EXPRESS:

```
*)
ENTITY revision;
  authorizing_documents :
     OPTIONAL LIST [1:?] OF other_reference_document;
  revision_approval : OPTIONAL person_and_organization;
  revision_date : OPTIONAL date;
  revision_description : OPTIONAL text;
  revision_level : identifier;
END_ENTITY;
(*
```

4.2.140.1 authorizing_documents

The authorizing_documents specifies the revision authorization documents when a revision description is not provided. The revision authorization documents may describe the revision history of the Tdp_element from its original release. The authorizing_documents may not be specified for a particular Revision. There may be more than one authorizing_documents for a Revision.

4.2.140.2 revision_approval

The revision_approval specifies the person and organization who are responsible for approving the revision. The revision_approval need not be specified for a particular Revision.

4.2.140.3 revision_date

The revision_date specifies the date and time that the revision_level was advanced. The revision_date need not be specified for a particular Revision.

EXAMPLE Figure 4 illustrates the format style for revision_date.

4.2.140.4 revision_description

The revision_description specifies a textual explanation of the revision. The revision_description need not be specified for a particular Revision.

4.2.140.5 revision_level

The revision_level specifies the version of the document from its original release or issue. Revisions reflect changes made to the original document after authorized release or issue that require the revision level to be advanced.

4.2.141 Security_classification

The Security_classification is company or government security or sensitivity parameters of the Tdp_element and its contents. The data associated with the Security_classification are the following:

— classification_date;

— classifier;

— declassification_date;

— item_classification;

— title_security_classification.

NOTE Application object EXPRESS:

```
*)
ENTITY security_classification;
  classification_date: OPTIONAL date;
  classifier : OPTIONAL person_and_organization;
  declassification_date: OPTIONAL date;
  item_classification : label;
  title_security_classification : OPTIONAL label;
END_ENTITY;
(*
```

4.2.141.1 classification_date

The classification_date specifies the date the item was given the item_classification. The classification_date need not be specified for a particular Security_classification.

EXAMPLE A radar absorbing material specification is classified on the date the specification is created.

4.2.141.2 classifier

The classifier specifies the organization or person in organization that defines or owns the classification. The classifier need not be specified for a particular Security_classification.

4.2.141.3 declassification_date

The declassification_date specifies the date the item was given the item_classification. The declassification_-date need not be specified for a particular security_classification.

EXAMPLE The secret performance specifications to a retired fighter jet is declassified on this date.

4.2.141.4 item_classification

The item_classification specifies the security level or sensitivity level of the Tdp_element. The security classification is assigned by the originator based on predetermined criteria.

4.2.141.5 title_security_classification

The title_security_classification specifies the security level or sensitivity of the title of the Tdp_element. The security classification is assigned by the originator based on predetermined Government or commercial

criteria restricting the title or nomenclature used within the title. The title_security_classification need not be specified for a particular Security_classification.

4.2.142 Sequence_effectivity

The Sequence_effectivity is a type of Effectivity (see 4.2.45) the intended use by a design authority of a Component within a range of Product_configurations (see 4.2.122) identified by serial numbers.

NOTE 1 The preceding definition is a paraphrased definition from ISO 10303-203 for Planned_sequence_ effectivity.

The data associated with a Sequence_effectivity are the following:

— from_effectivity_id;

— quantity_unit_of_measure;

— thru_effectivity_id;

— total_component_quantity_in_product_configuration.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY sequence_effectivity
  SUBTYPE OF (effectivity);
  from_effectivity_id : identifier;
  quantity_unit_of_measure : OPTIONAL measure;
  thru_effectivity_id : OPTIONAL identifier;
  total_component_quantity_in_product_configuration :
      OPTIONAL  quantity;
WHERE
  WR1: EXISTS (total_component_quantity_in_product_configuration)
    AND EXISTS (quantity_unit_of_measure);
END_ENTITY;
(*
```

4.2.142.1 from_effectivity_id

The from_effectivity_id specifies the beginning serial number of the range for the Sequence_effectivity.

NOTE The preceding definition is a paraphrased definition from ISO 10303-203 for Planned_ sequence_ effectivity.from_effectivity_id.

4.2.142.2 quantity_unit_of_measure

This quantity_unit_of_measure specifies the measure in terms that the total component_quantity_in_- product_configuration is expressed. The quantity_unit_of_measure need not be specified for a particular Sequence_effectivity.

NOTE The preceding definition is from ISO 10303-203 for quantity_unit_of_measure.

EXAMPLE Quantity_unit_of_measure may be rolls, sheets, or bars.

4.2.142.3 thru_effectivity_id

The thru_effectivity_id specifies the ending serial number of the range for the Sequence_effectivity. The thru_effectivity_id need not be specified for a particular Sequence_effectivity.

NOTE The preceding definition is a paraphrased definition from ISO 10303-203 for Planned_ sequence_ effectivity.thru_effectivity_id.

4.2.142.4 total_component_quantity_in_product_configuration

The total_component_quantity_in_product_configuration specifies the number of units that are effective for a particular part in a configuration. The total_component_quantity_in_product_configuration need not be specified for a particular Sequence_effectivity.

4.2.143 Shaded_shape_model

The Shaded_shape_model is a Shape_model (see 4.2.144) that has shading information applied to the model for presentation of the information. The data associated with a Shaded_shape_model are the following:

— part_shape_views.

NOTE Application object EXPRESS:

```
*)
ENTITY shaded_shape_model;
  part_shape_views : LIST [1:?] OF shape_model;
END_ENTITY;
(*
```

4.2.143.1 part_shape_views

The part_shape_views specify the viewing parameters for product shape presentation. There may be more than one part_shape_views for a Shaded_shape_model.

4.2.144 Shape_model

The Shape_model specifies the 3D or 2D mathematical representation of shape of the product model. The data associated with a Shape_model are the following:

— part_shape_representation;

— validation_properties.

NOTE Application object EXPRESS:

```
*)
ENTITY shape_model;
  part_shape_representation : geometry;
  validation_properties : OPTIONAL SET [1:?] OF
     geometric_validation_property;
END_ENTITY;
(*
```

4.2.144.1 part_shape_representation

The part_shape_representation specifies the underlying geometry that represents the Item shape.

4.2.144.2 validation_properties

The validation_properties specifies characteristics of a Shape_model that can be used to validate geometric translation results. There validation_properties need not be specified for a particular Shape_model. There may be more than one validation_properties for a Shape_model.

4.2.145 Sheet

The Sheet is a logical division of a Drawing (see 4.2.43) into a 2D area for the presentation of the product data. The divisions correspond to drawing size. A Sheet shall consist of at least one drawing view. The data associated with a Sheet are the following:

— presentation_format;

— sheet_configuration;

— sheet_identifier;

— sheet_size.

NOTE Application object EXPRESS:

```
*)
ENTITY sheet;
  presentation_format : OPTIONAL product_presentation;
  sheet_configuration: OPTIONAL configuration;
  sheet_identifier: OPTIONAL element_identification;
  sheet_size : OPTIONAL label;
WHERE
  WR1:EXISTS (sheet_configuration) OR EXISTS (sheet_identifier)
    OR EXISTS (sheet_size) OR EXISTS (presentation_format);
END_ENTITY;
(*
```

4.2.145.1 presentation_format

A presentation_format specifies the organization and composition of the drawing sheet for human visualization. The presentation_format need not be specified for a particular Sheet.

4.2.145.2 sheet_configuration

The sheet_configuration specifies the configuration management and data management characteristics about the Sheet. The sheet_configuration need not be specified for a particular Sheet.

4.2.145.3 sheet_identifier

The sheet_identifier specifies configuration management characteristics about the sheet that identify the sheet as a Tdp_element. The sheet_identifier need not be specified for a particular Sheet.

4.2.145.4 sheet_size

The sheet_size specifies a textual description of the size parameters of the drawing Sheet. The sheet_size need not be specified for a particular Sheet.

4.2.146 Simple_list_of_elements

The Simple_list_of_elements is a list of technical data package elements to be exchanged. The data associated with a Simple_list_of_elements are the following:

— element_entries.

NOTE Application object EXPRESS:

```
*)
ENTITY simple_list_of_elements;
  element_entries : LIST [1:?] OF data_definition_exchange_simple_entry;
END_ENTITY;
(*
```

4.2.146.1 element_entries

The element_entries specifies a list of Data_definition_simple_entry(s). There may be more than one element_entries for a Simple_list_of_elements.

4.2.147 Simple_list_of_files

The Simple_list_of_files is an ordered list of files exchanged. The data associated with a Simple_list_of_-files are the following:

— file_entries.

NOTE Application object EXPRESS:

```
*)
ENTITY simple_list_of_files;
  file_entries : LIST [1:?] OF data_definition_file_entry;
END_ENTITY;
(*
```

4.2.147.1 file_entries

The file_entries specifies an Accessed_file or Exchange_file that is exchanged. There may be more than one file_entries for a Simple_list_of_files.

EXAMPLE File “ABC.WPD” is at node 124.27.6.4 on the Internet.

NOTE Select type EXPRESS:

```
TYPE data_definition_file_entry = SELECT
  (accessed_file,
  exchange_file);
END_TYPE;
```

4.2.148 Single_document_list

A Single_document_list is a type of Document_list (see 4.2.41) that is either a Data_list (see 4.2.31) or an Index_list (see 4.2.85). A Single_document_list is limited to a single level of definition. Any Tdp_elements (see 4.2.170) referenced by sub-assemblies or component parts are not included.

NOTE 1 The Document_list shall contain a Data_list or an Index_list or both when the Indentured_ data_list is not available.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY single_document_list
  ABSTRACT SUPERTYPE OF (ONEOF(data_list,index_list))
  SUBTYPE OF (document_list);
END_ENTITY;
(*
```

4.2.149 Size_characteristics

The size_characteristics is a characterizing set of parameters about a Tdp_element (see 4.2.170) or File (see 4.2.59). Each Size_characteristics may be one of the following: Size_characteristics_sheet_across_file (see 4.2.152), Size_characteristics_full_size (see 4.2.150), or Size_characteristics_internal_divisions (see 4.2.151). The data associated with a Size_characteristics are the following:

— byte_size;

— sheet_size.

NOTE Application object EXPRESS:

```
*)
ENTITY size_characteristics SUPERTYPE OF
  (ONEOF (size_characteristics_sheet_across_file,
  size_characteristics_full_size,
  size_characteristics_internal_divisions));
  byte_size : OPTIONAL an_integer;
  sheet_size : OPTIONAL label;
WHERE
  WR1: EXISTS (byte_size) OR
  EXISTS (sheet_size);
END_ENTITY;
(*
```

4.2.149.1 byte_size

The byte_size specifies either the sum of the computer disk space required for all the files of one representation of a Tdp_element, or the computer disk space required for a particular File. The byte_size need not be specified for a particular Size_characteristics.

4.2.149.2 sheet_size

The sheet_size specifies the presentation size of a document page or sheet. The sheet_size need not be specified for a particular Size_characteristics.

4.2.150 Size_characteristics_full_size

A Size_characteristic_full_size is a type of Size_characteristics (see 4.2.149) that specifies that no additional information is required to specify the size information.

NOTE Application object EXPRESS:

```
*)
ENTITY size_characteristics_full_size
  SUBTYPE OF (size_characteristics);
END_ENTITY;
(*
```

4.2.151 Size_characteristics_internal_divisions

A Size_characteristic_internal_divisions is a type of Size_characteristics (see 4.2.149) that specifies that the File (see 4.2.59), document, or Tdp_element (see 4.2.170) has internal divisions that are sheets, views, pages or other internal division within the File, document, or Tdp_element.

The data associated with a Size_characteristics_internal_divisions are the following:

— internal_division_count;

— internal_division_type.

NOTE 1 Internal divisions are typically sheet, page, or view.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY size_characteristics_internal_divisions
  SUBTYPE OF (size_characteristics);
  internal_division_count : OPTIONAL an_integer;
  internal_division_type : OPTIONAL label;
WHERE
  WR1: EXISTS (internal_division_type)
       OR EXISTS (internal_division_count);
END_ENTITY;
(*
```

4.2.151.1 internal_division_count

The internal_division_count specifies the number of divisions identified in the internal_division_type. The internal_division_count need not be specified for a particular Size_characterstics_internal_divisions.

4.2.151.2 internal_division_type

The internal_division_type specifies that the method of classifying the internal division of the File, document, or Tdp_element. The internal_division_type need not be specified for a particular Size_-characterstics_ internal_divisions.

4.2.152 Size_characteristics_sheet_across_file

A Size_characteristics_sheet_across_file is a type of Size_characteristics (see 4.2.149) that specifies that a single sheet of a Tdp_element (see 4.2.170) is across one or more Files (see 4.2.59). The data associated with a Size_characteristics_ sheet_across_file are the following:

— frame_number_comprising_this_file;

— number_of_frames_for_sheet.

NOTE 1 The industrial practice of rasterizing frames of a document across several aperture cards resulted in a single sheet being across frames. Therefore, all of the frames had to be identified before a complete sheet could be assembled.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY size_characteristics_sheet_across_file
  SUBTYPE OF (size_characteristics);
  frame_number_comprising_this_file : OPTIONAL an_integer;
  number_of_frames_for_sheet : OPTIONAL an_integer;
WHERE
  WR1: EXISTS (number_of_frames_for_sheet) OR
  EXISTS (frame_number_comprising_this_file);
END_ENTITY;
(*
```

4.2.152.1 frame_number_comprising_this_file

The frame_number_comprising_this_file specifies the ordinal number for assembling the Files for a single sheet or page. The frames_number_comprising_this_file need not be specified for a particular Size_-characteristics_ sheet_across_file.

4.2.152.2 number_of_frames_for_sheet

The number_of_frames_for_sheet specifies the total number of Files that is required to assemble a single sheet or page of a Tdp_element. The number_of_frames_for_sheet need not be specified for a particular Size_characteristics_sheet_across_file.

4.2.153 Source_file

A Source_file is a type of File (see 4.2.59) that specifies that the File containing the required Tdp_element (see 4.2.170) or document.

NOTE Application object EXPRESS:

```
*)
ENTITY source_file
  SUBTYPE OF (file);
END_ENTITY;
(*
```

4.2.154 Source_information_type

A Source_information_type is identification of source of procurement for Items (see 4.2.89). The data associated with a Source_information_type are the following:

— source_code;

— source_description;

— type_of_coding_scheme.

NOTE Application object EXPRESS:

```
*)
ENTITY source_information_type;
  source_code : OPTIONAL label;
  source_description : OPTIONAL text;
  type_of_coding_scheme : OPTIONAL text;
WHERE
  WR1: EXISTS (source_code) OR
       EXISTS (source_description);
END_ENTITY;
(*
```

4.2.154.1 source_code

The source_code specifies mutually agreeable source procurement designations that a particular code defines. The source_code need not be specified for a particular Source_information_type.

4.2.154.2 source_description

The source_description specifies the description of a source_code. The source_description need not be specified for a particular Source_information_type.

4.2.154.3 type_of_coding_scheme

The type_of_coding_scheme specifies the coding scheme that is used for the procurement source information. The source_code_type_scheme identifies where the Item can be produced. The type_of_-coding_scheme need not be specified for a particular Source_information_type. The value of the type_of_-coding_scheme shall be a human_interpretable string of characters.

4.2.155 Source_location

A Source_location is a location on a computer system or a physical location of a thing. The data associated with a Source_information_type (see 4.2.154) are the following:

— path_information;

— storage_node_identification.

NOTE Application object EXPRESS:

```
*)
ENTITY source_location;
  path_information : text;
  storage_node_identification : text;
END_ENTITY;
(*
```

4.2.155.1 path_information

The path_information specifies the appropriate drive, directory, node or other location information required for the operating system software application environment within the network node or device or required for a person to physical location the item or document.

4.2.155.2 storage_node_identification

The storage_node_identification specifies the network protocol address of the network node or device or specifies the identifier of the phycial item or document relative to the path_information..

4.2.156 Special_condition

A Special_condition is a characteristic of an Item (see 4.2.89) or Tdp_element (see 4.2.170) that is mutually agreed to between parties for a business purpose. The data associated with the Special_condition are the following:

— code;

— description;

— type_of_coding_scheme.

NOTE Application object EXPRESS:

```
*)
ENTITY special_condition;
  code : OPTIONAL label;
  description : OPTIONAL text;
  type_of_coding_scheme : OPTIONAL text;
WHERE
  WR1: EXISTS (description) OR EXISTS (code);
END_ENTITY;
(*
```

4.2.156.1 code

The code specifies mutually agreeable special conditions characteristics that a particular code defines. The code need not be specified for a particular Special_condition.

4.2.156.2 description

The description specifies the text that characterizes the special condition. The description need not be specified for a particular Special_condition.

4.2.156.3 type_of_coding_scheme

The type_of_coding_scheme specifies the coding scheme that is used for the Special_conditions. The type_-of_coding_scheme need not be specified for a particular Special_condition.

NOTE Many companies and industries agree on different codes to depict different conditions. These coding scheme agreements are sometimes contained in industry or government standards.

The value of the type_of_coding_scheme shall be a human-interpretable string of characters.

EXAMPLE ANSI ASME Y14.34M is an example of a United States standard that contains a special_condition coding scheme.

4.2.157 Specification_document

A Specification_document is a type of Reference_document (see 4.2.135) that is a company, industry, government, commercial, or other specification.

NOTE Application object EXPRESS:

```
*)
ENTITY specification_document
  SUBTYPE OF (reference_document);
END_ENTITY;
(*
```

4.2.158 Specified_part_in_assembly_tree

A Specified_part_in_assembly_tree is the identification of an individual component in the assembly. The Specified_part_in_assembly_tree designates the existence of a higher level assembly and the constituent component. The data associated with the Specified_part_in_assembly_tree are the following:

— higher_assembly;

— sub_assembly.

NOTE 1 Specified_part_in_assembly_trees can be nested to identify specific instances of components in assembly structures with any number of levels.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY specified_part_in_assembly_tree
  SUBTYPE OF (assembly_relationship);
  higher_assembly : assembly_relationship;
  sub_assembly : quantified_part_usage_in_assembly;
UNIQUE
  UR1: higher_assembly, sub_assembly;
END_ENTITY;
(*
```

4.2.158.1 higher_assembly

The higher_assembly specifies an assembly that is a higher level parent of the Specified_part_in_-assembly_tree.

NOTE The higher_assembly specifies an Item_usage that is the same instance of the attribute component_of as this Specified_part_in_assembly_tree and the same instance of the attribute being_defined_for as the component_of of Quantified_part_usage_in_assembly referenced by the attribute sub_assembly.

4.2.158.2 sub_assembly

The description specifies the next lower level Item in the assembly tree.

NOTE The sub_assembly specifies a Quantified_part_usage_in_assembly that is the same instance of the attribute being_defined_for as this Specified_part_in_assembly_tree and the same instance of the item_usage referenced by the attribute component_of as the item_usage referenced by the attribute being_defined_for of the attribute higher_-assembly.

4.2.159 Standard_document

A Standard_document is a type of Reference_document (see 4.2.135) that is a company, industry, government, commercial, or other standard.

NOTE Application object EXPRESS:

```
*)
ENTITY standard_document
  SUBTYPE OF (reference_document);
END_ENTITY;
(*
```

4.2.160 Status

A Status is the identification of the process stage the information element is in during the life cycle of creating and supporting product information. The data associated with a Status are the following:

— status_code;

— status_code_basis.

EXAMPLE An action_status may be In-work, Complete, In-review, or Final.

NOTE Application object EXPRESS:

```
*)
ENTITY status;
  status_code : identifier;
  status_code_basis : OPTIONAL text;
END_ENTITY;
(*
```


4.2.160.1 status_code

The status_code specifies the process stage of the information.

NOTE Product Data Management (PDM) systems use this status to aid in identifying where in the life cycle an information element resides.

4.2.160.2 status_code_basis

The status_code_basis specifies the name of a particular set of process stages. The status_code_basis need not be specified for a particular Status.

NOTE A set of process stages may be standardized within a particular industry or among business partners.

4.2.161 Stock_material

A stock_material is raw material that material vendors supply for multiple uses. Stock_material usually comes in standard sizes based on material type. The data associated with a Stock_material are the following:

— material;

— size.

NOTE Application object EXPRESS:

```
*)
ENTITY stock_material;
  material : OPTIONAL item;
  size : OPTIONAL stock_size;
WHERE
  WR1: EXISTS (material) OR EXISTS (size);
END_ENTITY;
(*
```

4.2.161.1 material

The material specifies the base substance of the Stock_material. The material need not be specified for a particular Stock_material.

4.2.161.2 size

The size specifies the physical dimensions of the Stock_material. The size need not be specified for a particular Stock_material.

4.2.162 Stock_size

The Stock_size is the specification of the size, shape, and classification of the Stock_material (see 4.2.161) required for an Item (see 4.2.89). The data associated with a Stock_size are the following:

— catalog;

— cross_section;

— parameters_for.

NOTE Application object EXPRESS:

```
*)
ENTITY stock_size;
  parameters_for : stock_size_parameters;
  catalog : OPTIONAL stock_size_classification;
  cross_section : OPTIONAL stock_size_cross_section;
END_ENTITY;
(*
```

4.2.162.1 catalog

The catalog specifies the classification type for the Stock_size. The catalog need not be specified for a particular Stock_size. The value of the catalog shall be one of the following:

— bar;

— block;

— pipe;

— plate;

— roll;

— sheet_material;

— sphere;

— tube.

NOTE 1 See 4.2.162.1.1 - 4.2.162.1.8 for the definition of each allowable value for catalog.

NOTE 2 Application object EXPRESS:

```
*)
TYPE stock_size_classification = ENUMERATION OF
  (bar, plate, sheet_material, pipe, tube, block, sphere, roll);
END_TYPE;
(*
```


4.2.162.1.1 bar
A relatively long, straight, rigid piece of any solid material.

4.2.162.1.2 block
A solid piece of material having one or more flat sides.

4.2.162.1.3 pipe
Any hollow cylinder or tubular conveyance for a fluid or gas.

NOTE A pipe usually has thicker walls than something that is referred to as a tube.

4.2.162.1.4 plate
A smooth, flat, relatively thin, rigid body of uniform thickness.

4.2.162.1.5 roll
Material rolled up in the form of a cylinder.

EXAMPLE A roll of tape

4.2.162.1.6 sheet_material
A broad, thin, usually rectangular, material. A sheet can be non-rigid.

NOTE A non-rigid sheet could be laid over a curved surface.

4.2.162.1.7 sphere
a material where all points on its three-dimensional surface are equidistant from a fixed point.

4.2.162.1.8 tube
A hollow cylinder that conveys a fluid or functions as a passage.

NOTE A tube usually has thinner walls than something that is referred to as a pipe, A tube, because of its thinner walls, is something that is more likely to be bent to a specific functional shape from its stock shape.

4.2.162.2 cross_section

The cross_section specifies the form of the Stock_size when cut at right angles to an axis. The cross_section need not be specified for a particular Stock_size. The value of the cross_section shall be one of the following:

— angle;

— channel;

— hex;

— I_section;

— oval;

— rectangular;

— round;

— square;

— T_section.

NOTE 1 See 4.2.162.2.1 - 4.2.162.2.9 for the definition of each allowable value for cross_section.

NOTE 2 Application object EXPRESS:

```
*)
TYPE stock_size_cross_section = ENUMERATION OF
  (angle, channel, hex, I_s, oval, rectangular, round, square, T_s);
END_TYPE;
(*
```

4.2.162.2.1 angle
A cross section that represents a figure formed by two lines diverging from a common point that do not form a straight line.

4.2.162.2.2 channel
A cross section that represents a figure formed by three lines in a "U" shape.

4.2.162.2.3 hex
A cross section that represents a figure formed by six equal sides with all angles being equal.

4.2.162.2.4 I's
a cross section that represents a figure that looks like a capital "I".

4.2.162.2.5 oval
A cross section that represents a figure resembling an ellipse or formed by two half circles with two lines connecting the half circles at their end points tangentially.

4.2.162.2.6 rectangular
A cross section that represents a figure formed by a parallelogram with a right angle.

4.2.162.2.7 round
A cross section that represents a figure formed by a circle.

4.2.162.2.8 square
A cross section that represents a figure formed by a parallelogram with a right angle and equal length sides.

4.2.162.2.9 T's
A cross section that represents a figure that looks like a capital "T".

4.2.162.3 parameters_for

The parameters_for specify the measurement characteristics of size for the Stock_size. There may be more than one parameters_for for a Stock_size.

4.2.163 Stock_size_parameters

The Stock_size_parameters is the size characteristics of the Stock_material (see 4.2.161) required for an Item (see 4.2.89). The data associated with Stock_size_parameters are the following:

— a_length;

— diameter;

— height;

— name_of_parameter:

— numeric_parameter;

— radius;

— textual_parameter;

— wall_thickness;

— width.

NOTE 1 Figure 7 shows how these stock_size_parameters shall be applied to standard cross_sections . The key for Figure 7 is:

— D for diameter;

— H for height;

— R for radius;

— T for wall thickness;

— W for width.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY stock_size_parameters;
  a_length : OPTIONAL number_with_units;
  diameter : OPTIONAL number_with_units;
  height: OPTIONAL number_with_units;
  name_of_parameter : OPTIONAL label;
  numeric_parameter : OPTIONAL number_with_units;
  radius : OPTIONAL number_with_units;
```

```
    textual_parameter : OPTIONAL text;
    wall_thickness : OPTIONAL number_with_units;
    width : OPTIONAL number_with_units;
  WHERE
    WR1: EXISTS (textual_parameter) OR EXISTS (width) OR
         EXISTS (a_length) OR EXISTS (radius) OR EXISTS (height)
         OR EXISTS (diameter) OR EXISTS (wall_thickness)
         OR EXISTS (numeric_parameter);
  END_ENTITY;
  (*
```

4.2.163.1 a_length

The a_length specifies a measured distance or dimension for the extent of the stock. The a_length need not be specified for a particular Stock_size_parameters.

4.2.163.2 diameter

The diameter specifies the distances of a straight line through the center of a circular or spherical object. The diameter need not be specified for a particular Stock_size_parameters.

4.2.163.3 height

The height specifies the measured vertical extent of the stock. The height need not be specified for a particular Stock_size_parameters.

4.2.163.4 name_of_parameter

The name_of_parameter specifies a label for the stock size parameter. The name_of_parameter need not be specified for a particular Stock_size_parameter.

4.2.163.5 numeric_parameter

The numeric_parameter specifies a measured parameter of a stock material. The numeric_parameter need not be specified for a particular Stock_size_parameters.

NOTE The numeric_parameter allows business partners to determine additional stock size parameters.

4.2.163.6 radius

The radius specifies the measured distance or dimension of a straight line from the center of a circle or sphere to its circumference. The radius need not be specified for a particular Stock_size_parameters.

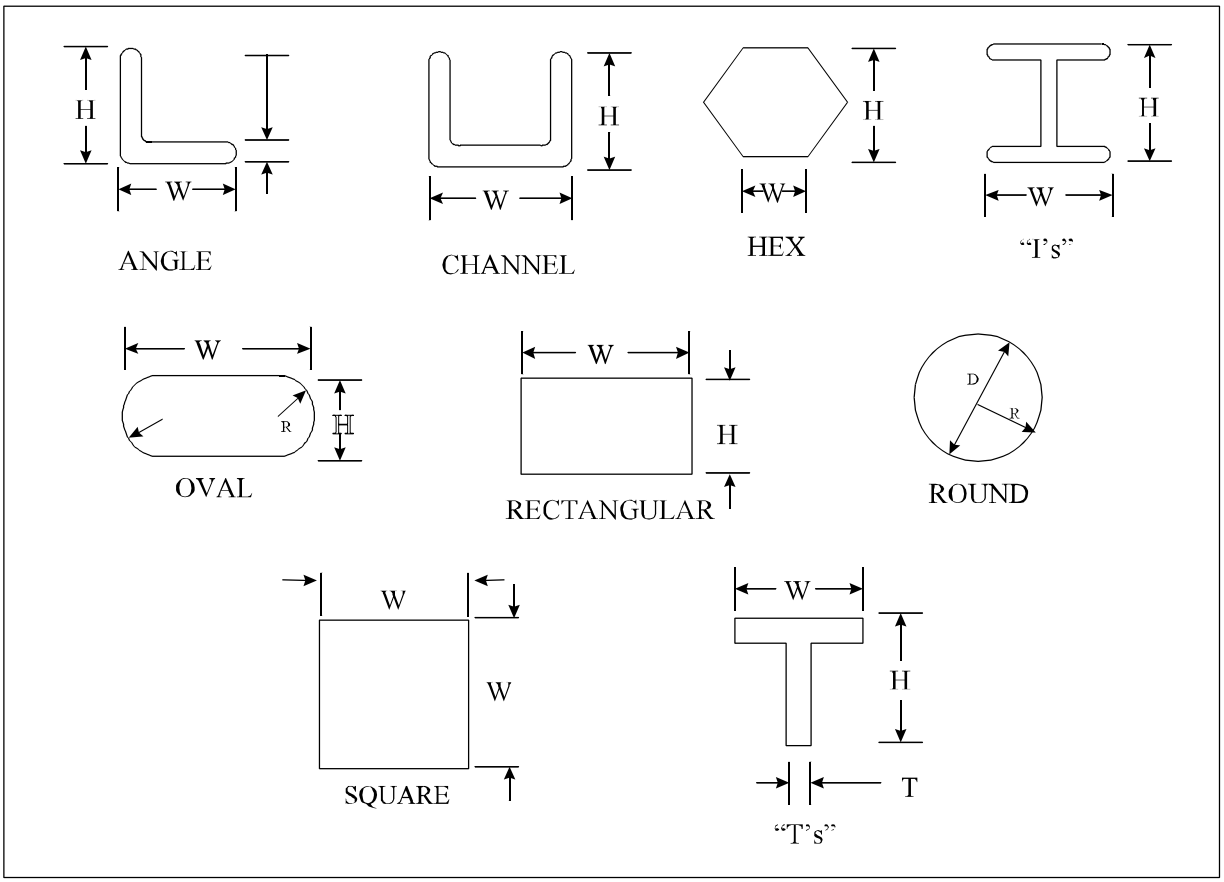

Figure 7 — Standard cross sections

4.2.163.7 textual_parameter

The textual_parameter specifies the size characteristics of the Stock_material in a text string. The textual_-parameter need not be specified for a particular Stock_size_parameters.

EXAMPLE 3" ID x 8 FT is the textual_parameters for a pipe used as Stock_material.

4.2.163.8 wall_thickness

The wall_thickness specifies the shortest measured distance or dimension between opposite surfaces. The wall_thickness need not be specified for a particular Stock_size_parameters.

NOTE The concept of thickness and height are considered equivalent.

EXAMPLE Thickness of the wall of a tube or pipe.

4.2.163.9 width

The width specifies the measured distance or dimension from side to side of an object.

4.2.164 String_location_definition

The String_location_definition is a type of Field_location (see 4.2.56) for the text location information for the title and content information. The data associated with a String_location_definition are the following:

— string_load_point.

NOTE Application object EXPRESS:

```
*)
ENTITY string_location_definition
SUBTYPE OF (field_location);
  string_load_point : load_point;
END_ENTITY;
(*
```

4.2.164.1 string_load_point

The string_load_point specifies the text location information for the string data. The Load_point specifies the bottom left corner of the first character of the text string.

4.2.165 System_declaration

The System_declaration is the identification of a computer system and the Tdp_element (see 4.2.170), File (see 4.2.59), or document of interest for that computer system. The data associated with a System_-declaration are the following:

— creating_interface;

— operating_system;

— system;

— system_element_identifier;

— system_related_element_identifier.

NOTE Application object EXPRESS:

```
*)
ENTITY system_declaration;
  creating_interface : OPTIONAL text;
  operating_system : OPTIONAL text;
  system_element_identifier : OPTIONAL identifier;
  system_related_element_identifier : OPTIONAL identifier;
  system : OPTIONAL identifier;
WHERE
     WR1: EXISTS (system_element_identifier) OR
```



```
      EXISTS (system_related_element_identifier) OR
      EXISTS (system) OR EXISTS(creating_interface) OR
      EXISTS (operating_system);
END_ENTITY;
(*
```

4.2.165.1 creating_interface

The creating_interface specifies the computer software that created the context file from the native format file. The creating_interface need not be specified for a particular System_declaration.

EXAMPLE CATIGE V418, CADDS - Plot Postscript printer driver, CATIA -STL.

4.2.165.2 operating_system

The operating_system specifies the computer software that manages the overall basic computer functions. The operating_system need not be specified for a particular System_declaration.

EXAMPLE AIX 4.1.2, Solaaris 5.2.1 Windows 98.

4.2.165.3 system

The system specifies the computer system of interest in the System_declaration. The system need not be specified for a particular System_declaration.

4.2.165.4 system_element_identifier

The system_element_identifier specifies identification of the File, document, or Tdp_ element of interest in the computer system. The system_element_identifier need not be specified for a particular System_-declaration.

4.2.165.5 system_related_element_identifier

The system_related_element_identifier specifies the File, document, or Tdp_element that the system_-element_identifier is related to. The system_related_element_identifier need not be specified for a particular System_declaration.

4.2.166 System_destination

The System_destination is the destination Company (see 4.2.17). The data associated with a System_-destination are the following:

— target_company;

— target_company_code;

— target_person.

NOTE Application object EXPRESS:

```
*)
ENTITY system_destination;
  target_company : company;
  target_company_code : OPTIONAL company_code;
  target_person : OPTIONAL person;
END_ENTITY;
(*
```

4.2.166.1 target_company

The target_company specifies the company for the System_destination.

4.2.166.2 target_company_code

The target_company_code specifies the unique code of the destination company. The target_company_code need not be specified for a particular System_destination.

4.2.166.3 target_person

The target_person specifies the destination person. The target_person need not be specified for a particular System_destination.

4.2.167 Tabulation

The tabulation is the table formatting information for table structures located in either the header or the body data. The data associated with a Tabulation are the following:

— group_parent_headings;

— parent_columns;

— tabulation_columnar_headings;

— tabulation_columns;

— tabulation_group_parent_headings;

— tabulation_headings.

NOTE Application object EXPRESS:

```
*)
ENTITY tabulation;
  group_parent_headings : OPTIONAL LIST [1:?] OF group_parent_header;
  parent_columns : OPTIONAL LIST [1:?] OF group_parent_content_holder;
  tabulation_columnar_headings : OPTIONAL LIST [1:?] OF column_header;
```



```
  tabulation_columns : LIST [1:?] OF columnar_data_content_holder;
  tabulation_group_parent_headings : OPTIONAL LIST [1:?] OF
  group_parent_column_header;
  tabulation_headings : OPTIONAL LIST [1:?] OF tabulation_header;
END_ENTITY;
(*
```

4.2.167.1 group_parent_headings

The group_parent_headings specifies the title text and its formatting and placement information for the group parent data. The group_parent_headings need not be specified for a particular Tabulation. There may be more than one group_parent_headings for a Tabulation.

4.2.167.2 tabulation_columnar_headings

The tabulation_columnar_headings specifies data content text and its formatting and placement information for the column headings in the table. The tabulation_columnar_headings need not be specified for a particular Tabulation. There may be more than one tabulation_columnar_headings for a Tabulation.

4.2.167.3 tabulation_columns

The tabulation_columns specifies data content text and its formatting and placement information for data in the table columns. The tabulation_columns need not be specified for a particular Tabulation. There may be more than one tabulation_columns for a Tabulation.

4.2.167.4 tabulation_group_parent_headings

The tabulation_group_parent_headings specifies the title text and its formatting and placement information for the group parent column header data. The tabulation_group_parent_headings need not be specified for a particular Tabulation. There may be more than one tabulation_group_parent_headings for a Tabulation.

4.2.167.5 tabulation_headings

The tabulation_headings specifies header title information for the table. This includes text, formatting, and placement information. The tabulation_headings need not be specified for a particular Tabulation. There may be more than one tabulation_headings for a Tabulation.

4.2.167.6 parent_columns

The parent_columns specifies the group parent data content text and its formatting and placement information. The parent_columns need not be specified for a particular Tabulation. There may be more than one parent_columns for a Tabulation.

4.2.168 Tabulation_entry

The Tabulation_entry is a row entry in an Other_list (see 4.2.109). Each Tabulation_entry contains a list of columnar entries for the row. The data associated with a Tabulation_entry are the following:

— column_information.

NOTE Application object EXPRESS:

```
*)
ENTITY tabulation_entry;
  column_information : LIST [1:?] OF text;
END_ENTITY;
(*
```

4.2.168.1 column_information

The column_information specifies the list of ordered text that pertains to each column in the tabulation for a specific row. There may be more than one column_information for a Tabulation_entry.

4.2.169 Tabulation_header

The tabulation_header is the information to define the table header text, style, and placement location of the title(s) for the table. The data associated with a Tabulation_header are the following:

— single_tabulation_header_titles.

NOTE Application object EXPRESS:

```
*)
ENTITY tabulation_header;
  single_tabulation_header_titles : LIST [1:?] OF field_title;
END_ENTITY;
(*
```

4.2.169.1 single_tabulation_header_titles

The single_tabulation_header_titles specifies the title text that points to the formatting and placement information for the text. There may be more than one single_tabulation_header_titles for a Tabulation_-header.

4.2.170 Tdp_element

A Tdp_element is any specific group of product data. A Tdp_element may be used as a portion of a design disclosure package. Each Tdp_element is either a Drawing (see 4.2.43), an Associated_list (see 4.2.11), a Product_data_set (see 4.2.123), or a Reference_document (see 4.2.135).

NOTE 1 The term Tdp_element used throughout this part of ISO 10303 identifies any specific group of product data over and above the four listed in the above definition.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY tdp_element
ABSTRACT SUPERTYPE OF (ONEOF(drawing,associated_list,
    product_data_set,reference_document));
END_ENTITY;
(*
```

4.2.171 Tdp_element_and_item_association

The Tdp_element_and_item_association is the relationship between an Item (see 4.2.89) and its Tdp_-element (see 4.2.170). The data associated with Tdp_element_and_item_association are the following:

— related_item;

— related_tdp_element.

NOTE Application object EXPRESS:

```
*)
ENTITY tdp_element_and_item_association;
  related_tdp_element : tdp_element;
  related_item : item;
END_ENTITY;
(*
```

4.2.171.1 related_item

The related_item specifies the Item that participates in the relationship.

4.2.171.2 related_tdp_element

The related_tdp_element specifies the Tdp_element that participates in the relationship.

4.2.172 Tdp_element_list_item

A Tdp_element_list_item is a Tdp_lement (see 4.2.170) identified in a Parts_list (see 4.2.117). The data associated with a Tdp_element_list_item are the following:

— reference_code;

— tdp_element_item.

NOTE 1 The type of list a tdp_element_list_item may apply to is a parts_list.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY tdp_element_list_item;
  reference_code : OPTIONAL identifier;
  tdp_element_item : element_identification;
END_ENTITY;
(*
```

4.2.172.1 reference_code

The reference_code specifies an identifier that is associated with the Tdp_element. The identifier is utilized to associate the Tdp_element with a note listed on the face of the drawing (that a Parts_list pertains), or a note listed and explained in another part of the body, or another listing source where detailing of notes are described. The reference_code may then be defined or explained in another portion of the Parts_list body or other specified listing source where detailing of notes are described. The reference_code need not be specified for a particular Tdp_element_list_item.

NOTE The reference_code may be used as a referencing method in lieu of describing a complete explanatory remark or notation directly. This may be done to compress output line requirements for presentation purposes or because of potentially repeating explanatory remarks and notations frequently. If the reference_code is used as a referencing method, then another Notation must exist that has the note attribute defined with the identical reference_code.

4.2.172.2 tdp_element_item

The tdp_element_item specifies the identification of the Tdp_element of interest.

4.2.173 Tdp_element_parent_to_item_child_relationship

A TDP_element_parent_to_item_child_relationship is a relationship that identifies that the child in the relationship is an Item (see 4.2.89) and that its parent is a Tdp_element (see 4.2.170).

The data associated with a Tdp_element_parent_to_item_child_relationship are the following:

— child;

— parent.

NOTE Application object EXPRESS:

```
*)
ENTITY tdp_element_parent_to_item_child_relationship;
  child : item_identification;
  parent : element_identification;
END_ENTITY;
(*
```


4.2.173.1 child

The child specifies the identification of the item of interest.

4.2.173.2 parent

The parent specifies the related parent Tdp_element of the child.

4.2.174 Tdp_element_parent_to_tdp_element_child_relationship

A Tdp_element_parent_to_tdp_element_child_relationship is a relationship that identifies that the child in the relationship is a Tdp_element (see 4.2.170) and that its parent is a Tdp_element. The data associated with a Tdp_element_parent_to_tdp_element_child_relationship are the following:

— child;

— parent.

NOTE Application object EXPRESS:

```
*)
ENTITY tdp_element_parent_to_tdp_element_child_relationship;
  child : element_identification;
  parent : element_identification;
END_ENTITY;
(*
```

4.2.174.1 child

The child specifies the identification of the Tdp_element of interest.

4.2.174.2 parent

The parent specifies the related parent Tdp_element.

4.2.175 Tdp_indentured_item

The Tdp_indentured_item is the identification of the top level Item (see 4.2.89) that head or start a part based top-down break-down listing. The data associated with a Tdp_indentured_item are the following:

— top_indenture.

NOTE 1 This entity enables the listing of multiple part based trees within the body of the list.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY tdp_indentured_item;
  top_indenture : item_identification;
END_ENTITY;
(*
```

4.2.175.1 top_indenture

The top_indenture specifies the top level Item in an indentured list.

4.2.176 Tdp_indentured_tdp_element

The Tdp_indentured_tdp_element is the identification of the top Tdp_elements (see 4.2.170) that head or start a document based document tree. The data associated with a Tdp_indentured_tdp_element are the following:

— top_indenture.

NOTE 1 This enables the listing of multiple document trees within the body of the list.

NOTE 2 Application object EXPRESS:

```
*)
ENTITY tdp_indentured_tdp_element;
  top_indenture : element_identification;
END_ENTITY;
(*
```

4.2.176.1 top_indenture

The tdp_indenture specifies the top level Tdp_element in an indentured list.

4.2.177 Text

The text is an alphanumeric string of characters that is intended to be human interpretable. It is for information and description purposes only.

NOTE Application object EXPRESS:

```
*)
TYPE text = STRING;
END_TYPE;
(*
```


4.2.178 Time

A Time specifies a moment of occurrence measured by hour, minute, and second. The data associated with a Time are the following:

— hour;

— minute;

— second;

— zone.

NOTE Application object EXPRESS:

```
*)
ENTITY time;
  hour: an_integer;
  minute: OPTIONAL an_integer;
  second: OPTIONAL a_real;
  zone: text;
END_ENTITY;
(*
```

4.2.178.1 hour

The hour specifies a specific period of time corresponding to one twenty-fourth of a day.

4.2.178.2 minute

The minute specifies a specific period of time corresponding to one sixtieth of a hour. The minute need not be specified for a particular Time.

4.2.178.3 second

The second specifies a specific period of time corresponding to one sixtieth of a minute. The second need not be specified for a particular Time.

4.2.178.4 zone

The zone specifies the relationship of the time to coordinated universal time.

4.2.179 Time_interval_effectivity

A Time_interval_effectivity is an Effectivity (see 4.2.45) that specifies the applicability of product data based on a span of time. The data associated with a Time_interval_effectivity are the following:

— duration;

— interval_name;

— primary_bound;

— secondary_bound.

NOTE Application object EXPRESS:

```
*)
ENTITY time_interval_effectivity
  SUBTYPE OF (effectivity);
  duration : OPTIONAL number_with_units;
  interval_name : label;
  primary_bound : date_or_event;
  secondary_bound : OPTIONAL date_or_event;
WHERE
  WR1: NOT (EXISTS (duration) AND EXISTS (secondary_bound));
END_ENTITY;
(*
```

4.2.179.1 duration

The duration specifies the length of the time interval. The duration need not be specified for a particular Time_interval_effectivity.

NOTE A duration's value can not exist if a secondary_bound has a value.

4.2.179.2 interval_name

The interval_name specifies words by which the interval is known.

4.2.179.3 primary_bound

The primary_bound specifies the start date or start event for the time interval.

NOTE Application object EXPRESS:

```
TYPE date_or_event = SELECT
 (date,
  event);
END_TYPE;
```

4.2.179.4 secondary_bound

The secondary_bound specifies the end date or end event for the time interval. A secondary_bound need not be specified for a particular Time_interval_effectivity.

NOTE 1 A secondary_bound's value can not exist if a duration has a value.

NOTE 2 Application object EXPRESS:

```
*)
TYPE date_or_event = SELECT
 (date,
  event);
END_TYPE;
(*
```

4.2.180 Transformation

The Transformation is a geometric placement and orientation composed of translation and rotation. Scaling is not include.

NOTE Application object EXPRESS:

```
*)
ENTITY transformation;
END_ENTITY;
(*
```

4.2.181 Volume

The Volume is the volumetric value of a component. Real numbers such as pounds and grams can be utilized for a Volume. The data associated with Volume are the following:

— a_value.

NOTE Application object EXPRESS:

```
*)
ENTITY volume;
  a_value: number_with_units;
END_ENTITY;
(*
```

4.2.181.1 a_value

The a_value specifies the volumetric quantity as a number with a unit of measure.

4.2.182 Weight

The Weight is the heaviness of the Item (see 4.2.89). The data associated with Weight are the following:

— a_value;

— derivation_method;

— unit_of_measure.

NOTE Application object EXPRESS:

```
*)
ENTITY weight;
  derivation_method : OPTIONAL weight_derivation;
  a_value : a_real;
  unit_of_measure : measure;
END_ENTITY;
(*
```

4.2.182.1 a_value

The a_value specifies the numerical value of the weight of the Item.

4.2.182.2 derivation_method

The derivation_method specifies the method that was utilized to determine the value of the Weight. The derivation_method need not be specified for a particular Weight. The value of derivation_method shall be one of the following:

— calculated;

— estimated;

— weighed.

NOTE 1 See 4.2.182.2.1 - 4.2.182.2.3 for the definition of each allowable value for derivation_method.

NOTE 2 Application object EXPRESS:

```
*)
TYPE weight_derivation = ENUMERATION OF
 (calculated,
  estimated,
  weighed);
END_TYPE;
(*
```

4.2.182.2.1 calculated
The weight was obtained using a numerical methodology.

4.2.182.2.2 estimated
The weight was obtained from best engineering estimate.

4.2.182.2.3 weighed
The weight of the Item was obtained through a measuring device.

4.2.182.3 unit_of_measure

The unit_of_measure specifies the standard measure units for the value attribute.

NOTE Application Object EXPRESS:

```
*)
END_SCHEMA;
(*
```

4.3 Application assertions

4.3.1 Accessed_file to Text

Each Accessed_file has storage_node_identification defined by zero or one Text objects. Each Text defines storage_node_identification for zero, one, or more Accessed_file objects.

NOTE 1 There are rules related to the Accessed_file object that requires the existence of at least one of the attributes.

Each Accessed_file has path_information defined by zero or one Text objects. Each Text has path_-information defined by zero, one, or more Accessed_file objects.

NOTE 2 There are rules related to the Accessed_file object that requires the existence of at least one of the attributes.

4.3.2 Alternate_element_identification to Change_identification

Each Alternate_element_identification has change_status defined by zero or one Change_identification objects. Each Change_identification defines change_status for zero, one, or more Alternate_element_-identification objects.

4.3.3 Alternate_element_identification to Design_authority

Each Alternate_element_identification has design_activity defined by exactly one Design_authority object. Each Design_authority defines design_activity for zero, one, or more Alternate_element_identification objects.

4.3.4 Alternate_element_identification to Element_type

Each Alternate_element_identification has element_classifications defined by one, or more Element_type objects. Each Element_type defines element_classifications for zero, one, or more Alternate_element_-identification objects.

4.3.5 Alternate_element_identification to Identifier

Each Alternate_element_identification has identifying_number defined by exactly one Identifier object. Each Identifer defines identifying_number for zero, one, or more Alternate_element_identification objects.

4.3.6 Alternate_element_identification to Other_reference_document

Each Alternate_element_identification has outstanding_changes defined by zero, one, or more Other_-reference_document objects. Each Other_reference_document defines outstanding_changes for zero, one, or more Alternate_element_identification objects.

4.3.7 Alternate_element_identification to Text

Each Alternate_element_identification has title defined by zero or one Text objects. Each Text defines title for zero, one, or more Alternate_element_identification objects.

4.3.8 Alternate_identification_of_item to Change_identification

Each Alternate_identification_of_item has change_status defined by zero or one Change_identification objects. Each Change_identification defines change_status for zero, one, or more Alternate_identification_-of_item objects.

4.3.9 Alternate_identification_of_item to Design_authority

Each Alternate_identification_of_item has design_activity defined by exactly one Design_authority object. Each Design_authority defines design_activity for zero, one, or many Alternate_identification_of_item object.

4.3.10 Alternate_identification_of_item to Drawing_suffix_number_-combination

Each Alternate_identification_of_item has identifying_number defined by zero or one Drawing_suffix_-number_combination objects. Each Drawing_suffix_number_combination defines identifying_number for zero, one, or more Alternate_identification_of_item objects.

NOTE This assertion is established through item_identifying_number_select.

4.3.11 Alternate_identification_of_item to Identifier

Each Alternate_identification_of_item has identifying_number defined by zero or one Identifier objects. Each Identifier defines identifying_number for zero, one, or more Alternate_identification_of_item objects.

NOTE This assertion is established through item_identifying_number_select.

4.3.12 Alternate_identification_of_item to Label

Each Alternate_identification_of_item has classifications defined by zero, one, or more Label objects. Each Label defines classifications for zero, one, or more Alternate_identification_of_item objects.

Each Alternate_identification_of_item has nomenclature_or_name defined by zero or one Label objects. Each Label defines nomenclature_or_name for zero, one, or more Alternate_identification_of_item objects.

4.3.13 Alternate_identification_of_item to Source_information_type

Each Alternate_identification_of_item has source_information defined by zero or one Source_information_-type objects. Each Source_information_type defines nomenclature_or_name for zero, one, or more Alternate_identification_of_item objects.

4.3.14 Alternate_item to An_Integer

Each Alternate_item has preference_order defined by zero or one An_integer objects. Each An_integer defines preference_order for zero, one, or more Alternate_item objects.

4.3.15 Alternate_item to Item

Each Alternate_item has interchange_item defined by zero or one Item object. Each Item defines interchange_item for zero, one, or more Alternate_item objects.

NOTE This assertion is established through usage_context.

4.3.16 Alternate_item to Item_usage

Each Alternate_item has interchange_item defined by zero or one Item_usage object. Each Item_usage defines interchange_item for zero, one, or more Alternate_item objects.

NOTE This assertion is established through usage_context.

4.3.17 Alternate_item to Label

Each Alternate_item has type_of_alternate defined by zero or one Label objects. Each Label defines type_-of_alternate for zero, one, or more Alternate_item objects.

4.3.18 Alternate_item to Text

Each Alternate_item has usage_conditions defined by zero, one or more Text objects. Each Text defines usage_conditions for zero, one, or more Alternate_item objects.

4.3.19 Annotation to Geometry

Each Annotation has annotation_geometry defined by zero or one Geometry objects. Each Geometry defines annotation_geometry for zero, one, or more Annotation objects.

4.3.20 Approval to Date

Each Approval has approval_date defined by exactly one Date object. Each Date defines date for zero, one, or more Approval objects.

4.3.21 Approval to Person_and_organization

Each Approval has person_organization defined by exactly one Person_and_organization object. Each Person_and_organization defines person_organization for zero, one, or more Approval objects.

4.3.22 Approval to Text

Each Approval has purpose_of defined by exactly one Text object. Each Text defines purpose_of for zero, one, or more Approval objects.

Each Approval has status defined by exactly one Text object. Each Text defines status for zero, one, or more Approval objects.

4.3.23 Associated_list to Drawing

Each Associated_list has related_to defined by zero or one Drawing objects. Each Drawing defines related_-to for zero, one, or more Associated_list objects.

4.3.24 Associated_list to Product_data_set

Each Associated_list has related_to defined by zero or one Product_data_set objects. Each Product_data_set defines related_to for zero, one or more Associated_list objects.

4.3.25 Associated_list to List_presentation

Each Associated_list has presentation defined by zero or one List_presentation objects. Each List_-presentation defines presentation for zero, one, or more Associated_list objects.

4.3.26 Body_presentation to Tabulation

Each Body_presentation has body_tabulation_presentation defined by one or more Tabulation objects. Each Tabulation defines body_tabulation_presentation for zero, one, or more Body_presentation objects.

4.3.27 Certification to Approval

Each Certification has approvals defined by zero, one or more Approval objects. Each Approval defines approvals for zero, one, or more Certification objects.

4.3.28 Certification to Date

Each Certification has creation_date defined by zero or one Date objects. Each Date defines creation_date for zero, one, or more Certification objects.

4.3.29 Certification to Label

Each Certification has name defined by exactly one Label object. Each Label defines name for zero, one, or more Certification objects.

4.3.30 Certification to Text

Each Certification has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Certification objects.

4.3.31 Change_identification to Approval

Each Change_identification has revision_authorization_identifications defined by zero, one, or more Approval objects. Each Approval defines revision_authorization_identifications for zero, one, or more Change_identification objects.

NOTE This assertion is established through revision_authorization_select.

4.3.32 Change_identification to Change_type

Each Change_identification has change_code defined by zero or one Change_type objects. Each Change_-type defines change_code for zero, one, or more Change_identification objects.

4.3.33 Change_identification to Contract

Each Change_identification has revision_authorization_identifications defined by zero or one Contract objects. Each Contract defines revision_authorization_identifications for zero, one, or more Change_-identification objects.

NOTE This assertion is established through revision_authorization_select.

4.3.34 Change_identification to Date

Each Change_identification has change_date defined by zero or one Date objects. Each Date defines change_date for zero, one, or more Change_identification objects.

NOTE 1 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

Each Change_identification has issue_date defined by zero or one Date objects. Each Date defines issue_-date for zero, one, or more Change_identification objects.

NOTE 2 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

Each Change_identification has revision_date defined by zero or one Date objects. Each Date defines revision_date for zero, one, or more Change_identification objects.

NOTE 3 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

4.3.35 Change_identification to Identifier

Each Change_identification has change_level defined by zero or one Identifier objects. Each Identifier defines change_level for zero, one, or more Change_identification objects.

NOTE 1 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

Each Change_identification has issue_level defined by zero or one Identifier objects. Each Identifier defines issue_level for zero, one, or more Change_identification objects.

NOTE 2 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

Each Change_identification has revision_level defined by zero or one Identifier objects. Each Identifier defines revision_level for zero, one, or more Change_identification objects.

NOTE 3 There are rules related to the Change_identification object that requires the existence of at least one of the attributes.

Each Change_identification has revision_authorization_identifications defined by zero, one, or more Identifier objects. Each Identifier defines revision_authorization_identifications for zero, one or more Change_identification objects.

NOTE 4 This assertion is established through revision_authorization_select.

4.3.36 Change_identification to Other_reference_document

Each Change_identification has revision_authorization_identifications defined by zero, one, or more Other_-reference_document objects. Each Other_reference_document defines revision_authorization_identifications for zero, one, or more Change_identification objects.

NOTE This assertion is established through revision_authorization_select.

4.3.37 Change_identification to Person_and_organization

Each Change_identification has assigned_to defined by zero, one or more Person_and_organization objects. Each person_and_organization defines assigned_to for zero, one, or more Change_identification objects.

4.3.38 Change_identification to Status

Each Change_identification has action_status defined by zero or one Status objects. Each Status defines action_status for exactly one Change_identification objects.

4.3.39 Change_identification to Text

Each Change_identification has change_description defined by zero or one Text objects. Each Text defines change_description for zero, one, or more Change_identification objects.

Each Change_identification has revision_description defined by zero or one Text objects. Each Text defines revision_description for zero, one, or more Change_identification objects.

4.3.40 Column_header to Field_title

Each Column_header has single_column_header_titles defined by one or more Field_title objects. Each Field_title defines single_column_header_titles for zero, one, or more Column_header objects.

4.3.41 Columnar_data_content_holder to Field_content_holder_tabulation

Each Columnar_data_content_holder has column_content_presentation defined by exactly one Field_-content_ holder_tabulation object. Each Field_content_holder_tabulation defines column_content_-presentation for zero, one, or more Columnar_data_content_holder objects.

4.3.42 Company to Label

Each Company has name defined by exactly one Label object. Each Label defines name for zero, one, or more Company objects.

4.3.43 Company to Text

Each Company has address defined by zero or one Text objects. Each Text defines address for zero, one, or more Company objects.

4.3.44 Company_code to Company

Each Company_code has code_administrator defined by zero or one Company objects. Each Company defines code_administrator for zero, one, or more Company_code objects.

4.3.45 Company_code to Identifier

Each Company_code has code defined by exactly one Identifier object. Each Identifier defines code for zero, one, or many Company_code objects.

4.3.46 Company_code to Text

Each Company_code has type_of_code defined by exactly one Text object. Each Text defines type_of_code for zero, one, or more Company_code objects.

4.3.47 Configuration to Contract

Each Configuration has preparing_contracts defined by zero, one, or more Contract objects. Each Contract defines preparing_contracts for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.48 Configuration to Distribution_notice

Each Configuration has distribution_authorizations defined by zero, one, or more Distribution_notice objects. Each Distribution_notice defines distribution_authorizations for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.49 Configuration to Label

Each configuration has data_usage_rights defined by zero, or one Label. Each Label defines data_usage_-rights for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.50 Configuration to Product_configuration

Each Configuration has end_item_system_designation defined by zero or one Product_configuration objects. Each Product_configuration defines end_item_system_designation for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.51 Configuration to Release_authentication

Each Configuration has release_authorizations defined by zero, one, or more Release_ authentication objects. Each Release_authentication defines release_authorization for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.52 Configuration to Security_classification

Each Configuration has security_identifications defined by zero, one, or more Security_classification objects. Each Security_classification defines security_identifications for zero, one, or more Configuration objects.

NOTE There are rules related to the Configuration object that requires the existence of at least one of the attributes.

4.3.53 Content_property to A_real

Each Content_property has real_world_scale defined by zero or one A_real objects. Each A_real defines real_world_scale for zero, one, or more Content_property objects.

4.3.54 Content_property to Text

Each Content_property has detail_level defined by zero or one Text objects. Each Text defines detail_level for zero, one, or more Content_property objects.

Each Content_property has geometry_type defined by zero or one Text objects. Each Text defines geometry_type for zero, one, or more Content_property objects.

Each Content_property has languages defined by zero, one or more Text objects. Each Text defines languages for zero, one, or more Content_property objects.

4.3.55 Contract to Approval

Each Contract has approvals defined by zero, one or more Approval objects. Each Approval defines approvals for zero, one, or more Contract objects.

4.3.56 Contract to Date

Each Contract has creation_date defined by zero or one Date objects. Each Date defines creation_date for zero, one, or more Contract objects.

4.3.57 Contract to Identifier

Each Contract has contract_number defined by exactly one Identifier object. Each Identifier defines contract_number for zero, one, or more Contract objects.

Each Contract has contract_data_requirements_list defined by zero or one Identifier objects. Each Identifier defines contract_data_requirements_list for zero, one, or more Contract objects.

Each Contract has data_item_description_identification defined by zero or one Identifier objects. Each Identifier defines data_item_description_identification for zero, one, or more Contract objects.

4.3.58 Contract to Person_and_organization

Each Contract has affected_organizations defined by zero, one or more Person_and_organization objects. Each Person_and_organization defines affected_organizations for zero, one, or more Contract objects.

4.3.59 Contract_submission to Contract

Each Contract_submission has delivered_contract defined by exactly one Contract object. Each Contract defines delivered_contract for zero, one, or more Contract_submission objects.

4.3.60 Contract_submission to Date

Each Contract_submission has date_of_submission defined by exactly one Date object. Each Date defines date_of_submission for zero, one, or more Contract_submission objects.

4.3.61 Contract_submission to Text

Each Contract_submission has location defined by exactly one Text object. Each Text defines location for zero, one, or more Contract_submission objects.

4.3.62 Data_definition_entry_item to Accessed_file

Each Data_definition_entry_item has entry_files defined by zero, one, or more Accessed_file objects. Each Accessed_file defines entry_files for zero, one, or more Data_definition_entry_item objects.

NOTE This assertion is established through data_definition_file_entry.

4.3.63 Data_definition_entry_item to Contract_submission

Each Data_definition_entry_item has available_from defined by zero, one or more Contract_submission objects. Each Contract_submission defines available_from for zero, one or more Data_definition_entry_item objects.

NOTE This assertion is established through source_identification.

4.3.64 Data_definition_entry_item to Delivery_accounting

Each Data_definition_entry_item has delivery_accounting_references defined by zero, one or more Delivery_accounting objects. Each Delivery_accounting defines delivery_accounting_references for zero, one or more Data_definition_entry_item objects.

4.3.65 Data_definition_entry_item to Exchange_file

Each Data_definition_entry_item has entry_files defined by zero, one or more Exchange_file objects. Each Exchange_file defines entry_files for zero, one, or many Data_definition_entry_item objects.

NOTE This assertion is established through data_definition_file_entry.

4.3.66 Data_definition_entry_item to Identifier

Each Data_definition_entry_item has entry_item_change_level defined by zero, one, or more Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more Data_definition_entry_item objects.

4.3.67 Data_definition_entry_item to Item_identification

Each Data_definition_entry_item has superseded_entry defined by zero or one Item_identification objects. Each Item_identification defines superseded_entry for zero, one, or more Data_definition_entry_item objects.

4.3.68 Data_definition_entry_item to Notation

Each Data_definition_entry_item has entry_notes defined by zero, one, or more Notation objects. Each Notation defines entry_notes for zero, one, or more Data_definition_entry_item objects.

4.3.69 Data_definition_entry_item to Reference_document

Each Data_definition_entry_item has available_from defined by zero, one, or more Reference_document objects. Each Reference_document defines available_from for zero, one, or more Data_definition_entry_item objects.

NOTE This assertion is established through source_identification.

4.3.70 Data_definition_entry_item to Source_location

Each Data_definition_entry_item has available_from defined by zero, one, or more Source_location objects. Each Source_location defines available_from for zero, one, or more Data_definition_entry_item objects.

NOTE This assertion is established through source_identification.

4.3.71 Data_definition_entry_item to Special_condition

Each Data_definition_entry_item has special_conditions defined by zero, one, or more Special_condition objects. Each Special_condition defines special_conditions for zero, one, or more Data_definition_entry_-item objects.

4.3.72 Data_definition_entry_tdp_element to Accessed_file

Each Data_definition_entry_tdp_element has entry_files defined by zero, one, or more Accessed_file objects. Each Accessed_file defines entry_files for zero, one, or more Data_definition_entry_tdp_element objects.

NOTE This assertion is established through data_definition_file_entry.

4.3.73 Data_definition_entry_tdp_element to Content_property

Each Data_definition_entry_tdp_element has entry_content_property defined by zero or one Content_-property objects. Each Content_property defines entry_content_property for zero, one or more Data_-definition_entry_tdp_element objects.

4.3.74 Data_definition_entry_tdp_element to Contract_submission

Each Data_definition_entry_tdp_element has available_from defined by zero, one or more Contract_-submission objects. Each Contract_submission defines available_from for zero, one or more Data_-definition_entry_tdp_element objects.

NOTE This assertion is established through source_identification.

4.3.75 Data_definition_entry_tdp_element to Delivery_accounting

Each Data_definition_entry_tdp_element has delivery_accounting_references defined by zero, one or more Delivery_accounting objects. Each Delivery_accounting defines delivery_accounting_references for zero, one or more Data_definition_entry_tdp_element objects.

4.3.76 Data_definition_entry_tdp_element to Element_identification

Each Data_definition_entry_tdp_element has superseded_entry defined by zero or one Element_-identification objects. Each Element_identification defines superseded_entry for zero, one, or more Data_-definition_entry_tdp_element objects.

4.3.77 Data_definition_entry_tdp_element to Exchange_file

Each Data_definition_entry_tdp_element has entry_files defined by zero, one or more Exchange_file objects. Each Exchange_file defines entry_files for zero, one, or many Data_definition_entry_tdp_element objects.

NOTE This assertion is established through data_definition_file_entry.

4.3.78 Data_definition_entry_tdp_element to File_format

Each Data_definition_entry_tdp_element has entry_format defined by zero or one File_format objects. Each File_format defines entry_format for zero, one, or more Data_definition_entry_tdp_element objects.

NOTE This assertion is established through entry_format_select.

4.3.79 Data_definition_entry_tdp_element to Identifier

Each Data_definition_entry_tdp_element has entry_item_change_level defined by zero, one, or more Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more Data_definition_-entry_tdp_element objects.

4.3.80 Data_definition_entry_tdp_element to Label

Each Data_definition_entry_tdp_element has actual_inclusion_in_data_exchange defined by zero, one, or more Label objects. Each Label defines actual_inclusion_in_data_exchange for zero, one, or more Data_-definition_entry_tdp_element objects. Each Data_definition_entry_tdp_element has data_usage_rights defined by zero or one Label objects. Each Label defines data_usage_rights for zero, one, or more Data_-definition_entry_tdp_element objects.

4.3.81 Data_definition_entry_tdp_element to Notation

Each Data_definition_entry_tdp_element has entry_notes defined by zero, one, or more Notation objects. Each Notation defines entry_notes for zero, one, or more Data_definition_entry_tdp_element objects.

4.3.82 Data_definition_entry_tdp_element to Reference_document

Each Data_definition_entry_tdp_element has available_from defined by zero, one, or more Reference_-document objects. Each Reference_document defines available_from for zero, one, or more Data_-definition_entry_tdp_element objects.

NOTE This assertion is established through source_identification.

4.3.83 Data_definition_entry_tdp_element to Size_characteristics

Each Data_definition_entry_tdp_element has size defined by zero or one Size_characteristics objects. Each Size_characteristics defines size for zero, one, or more Data_definition_entry_tdp_element objects.

4.3.84 Data_definition_entry_tdp_element to Source_location

Each Data_definition_entry_tdp_element has available_from defined by zero, one, or more Source_location objects. Each Source_location defines available_from for zero, one, or more Data_definition_entry_tdp_-element objects.

NOTE This assertion is established through source_identification.

4.3.85 Data_definition_entry_tdp_element to Special_condition

Each Data_definition_entry_tdp_element has special_conditions defined by zero, one, or more Special_-condition objects. Each Special_condition defines special_conditions for zero, one, or more Data_-definition_entry_tdp_element objects.

4.3.86 Data_definition_entry_tdp_element to Text

Each Data_definition_entry_tdp_element has entry_format defined by zero or one Text objects. Each Text defines entry_format for zero, one, or more Data_definition_entry_tdp_element objects.

NOTE This assertion is established through entry_format_select.

Each Data_definition_entry_tdp_element has master_file defined by zero or one Text objects. Each Text defines master_file for zero, one, or more Data_definition_entry_tdp_element objects.

4.3.87 Data_definition_entry_tdp_element to Yes_no

Each Data_definition_entry_tdp_element has master_file defined by zero or one Yes_no objects. Each Yes_-no defines master_file for zero, one, or more Data_definition_entry_tdp_element objects.

4.3.88 Data_definition_exchange to Data_definition_exchange_body

Each Data_definition_exchange has list_body defined by exactly one Data_definition_exchange_body object. Each Data_definition_exchange_body defines list_body for exactly one Data_definition_exchange objects.

4.3.89 Data_definition_exchange to Data_definition_exchange_header

Each Data_definition_exchange has list_header defined by exactly one Data_definition_exchange_header object. Each Data_definition_exchange_header defines list_header for exactly one Data_definition_exchange objects.

4.3.90 Data_definition_exchange to List_presentation

Each Data_definition_exchange has presentation defined by zero or one List_presentation objects. Each List_presentation defines presentation for zero, one, or more Data_definition_exchange objects.

4.3.91 Data_definition_exchange_body to Data_definition_indentured_list_-method

Each Data_definition_exchange_body has list_method defined by exactly one Data_definition_indentured_-list_method object. Each Data_definition_indentured_list_method defines list_method for zero, one, or more Data_definition_exchange_body objects.

NOTE This assertion is established through data_definition_exchange_list_method.

4.3.92 Data_definition_exchange_body to Notation

Each Data_definition_exchange_body has notes_list defined by zero, one, or more Notation objects. Each Notation defines notes_list for zero, one, or more Data_definition_exchange_body objects.

4.3.93 Data_definition_exchange_body to Revision

Each Data_definition_exchange_body has revision_history defined by zero, one, or more Revision objects. Each Revision defines revision_history for by zero, one, or more Data_definition_exchange_body objects.

4.3.94 Data_definition_exchange_body to Simple_list_of_elements

Each Data_definition_exchange_body has list_method defined by exactly one Simple_list_of_elements object. Each Simple_list_of_elements defines list_method for zero, one, or more Data_definition_exchange_-body objects.

NOTE This assertion is established through data_definition_exchange_list_method.

4.3.95 Data_definition_exchange_body to Simple_list_of_files

Each Data_definition_exchange_body has list_method defined by exactly one Simple_list_of_files object. Each Simple_list_of_files defines list_method for zero, one, or more Data_definition_exchange_body objects.

NOTE This assertion is established through data_definition_exchange_list_method.

4.3.96 Data_definition_exchange_header to Date

Each Data_definition_exchange_header has date_of_transfer defined by zero or one Date objects. Each Date defines date_of_transfer for zero, one, or many Data_definition_exchange_header objects.

4.3.97 Data_definition_exchange_header to Delivery_accounting

Each Data_definition_exchange_header has delivery_accounting_references defined by zero, one, or more Delivery_accounting objects. Each Delivery_accounting defines delivery_accounting_references for zero, one, or more Data_definition_exchange_header objects.

4.3.98 Data_definition_exchange_header to Element_identification

Each Data_definition_exchange_header has procurement_references defined by zero, one, or more Element_identification objects. Each Element_identification defines procurement_references for zero, one, or more Data_definition_exchange_header objects.

NOTE This assertion is established through identification_select.

4.3.99 Data_definition_exchange_header to Item_identification

Each Data_definition_exchange_header has procurement_references defined by zero, one, or more Item_-identification objects. Each Item_identification defines procurement_references for zero, one, or more Data_-definition_exchange_header objects.

NOTE This assertion is established through identification_select.

4.3.100 Data_definition_exchange_header to Header

Each Data_definition_exchange_header has common_header defined by exactly one Header object. Each Header defines common_header for zero or one Data_definition_exchange_header objects.

4.3.101 Data_definition_exchange_header to Reason

Each Data_definition_exchange_header has exchange_reason defined by exactly one Reason object. Each Reason defines exchange_reason for zero, one, or more Data_definition_exchange_header objects.

4.3.102 Data_definition_exchange_header to System_destination

Each Data_definition_exchange_header has destinations defined by zero, one, or more System_destination objects. Each System_destination defines destinations for zero, one, or more Data_definition_exchange_-header objects.

4.3.103 Data_definition_exchange_simple_entry to Data_definition_entry_tdp_-element

Each Data_definition_exchange_simple_entry has entry defined by exactly one Data_definition_entry_tdp_-element object. Each Data_definition_entry_tdp_element defines entry for zero, one, or more Data_-definition_exchange_simple_entry objects.

4.3.104 Data_definition_exchange_simple_entry to Element_identification

Each Data_definition_exchange_simple_entry has simple_entry defined by exactly one Element_-identification object. Each Element_identification defines simple_entry for zero, one, or more Data_-definition_exchange_simple_entry objects.

4.3.105 Data_definition_indentured_entry to Data_definition_entry_item

Each Data_definition_indentured_entry has entry_characteristics defined by exactly one Data_definition_-entry_item object. Each Data_definition_entry_item defines entry_characteristics for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through data_definition_entry_select.

4.3.106 Data_definition_indentured_entry to Data_definition_entry_tdp_-element

Each Data_definition_indentured_entry has entry_characteristics defined by exactly one Data_definition_-entry_tdp_element object. Each Data_definition_entry_tdp_element defines entry_characteristics for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through data_definition_entry_select.

4.3.107 Data_definition_indentured_entry to Effectivity

Each Data_definition_indentured_entry has effective_on defined by zero or one Effectivity objects. Each Effectivity defines effective_on for zero, one, or more Data_definition_indentured_entry objects.

4.3.108 Data_definition_indentured_entry to Item_parent_to_item_child_-relationship

Each Data_definition_indentured_entry has indentured_entry defined by exactly one item_parent_to_item_-child_relationship object. Each item_parent_to_item_child_relationship defines indentured_entry for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.109 Data_definition_indentured_entry to Item_parent_to_tdp_element_-child_relationship

Each Data_definition_indentured_entry has indentured_entry defined by exactly one item_parent_to_tdp_-element_child_relationship object. Each item_parent_to_tdp_element_child_relationship defines indentured_-entry for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.110 Data_definition_indentured_entry to Label

Each Data_definition_indentured_entry has indenture_level defined by zero or one Label objects. Each Label defines indenture_level for zero, one, or more Data_definition_indentured_entry objects.

4.3.111 Data_definition_indentured_entry to Tdp_element_parent_to_item_-child_relationship

Each Data_definition_indentured_entry has indentured_entry defined by exactly one tdp_element_parent_-to_item_child_relationship object. Each tdp_element_parent_to_item_child_relationship defines indentured_-entry for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.112 Data_definition_indentured_entry to Tdp_element_parent_to_tdp_-element_child_relationship

Each Data_definition_indentured_entry has indentured_entry defined by exactly one tdp_element_parent_-to_tdp_element_child_relationship object. Each tdp_element_parent_to_tdp_element_child_relationship defines indentured_entry for zero, one, or more Data_definition_indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.113 Data_definition_indentured_entry to Tdp_indentured_item

Each Data_definition_indentured_entry has indentured_entry defined by exactly one tdp_indentured_item object. Each tdp_indentured_item defines indentured_entry for zero, one, or more Data_definition_-indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.114 Data_definition_indentured_entry to Tdp_indentured_tdp_element

Each Data_definition_indentured_entry has indentured_entry defined by exactly one tdp_indentured_tdp_-element object. Each tdp_indentured_tdp_element defines indentured_entry for zero, one, or more Data_-definition_indentured_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.115 Data_definition_indentured_list_method to Data_definition_-indentured_entry

Each Data_definition_indentured_list_method has order_of defined by one or more Data_definition_-indentured_entry objects. Each Data_definition_indentured_entry defines order_of defined by zero, one, or more Data_definition_indentured_list_method objects.

4.3.116 Data_definition_indentured_list_method to Indentured_list_by_-document

Each Data_definition_indentured_list_by_document has method_of_entry_tabulation defined by exactly one Indentured_list_method object. Each Indentured_list_method defines method_of_entry for zero, one, or more Data_definition_indentured_list_by_document objects.

NOTE This assertion is established through indentured_list_method.

4.3.117 Data_definition_indentured_list_method to Indentured_list_by_part

Each Data_definition_indentured_list_by_part has method_of_entry_tabulation defined by exactly one Indentured_list_method object. Each Indentured_list_method defines method_of_entry for zero, one, or more Data_definition_indentured_list_by_part objects.

NOTE This assertion is established through indentured_list_method.

4.3.118 Data_definition_indentured_list_method to Indentured_list_by_part_-with_document_references_to_parts

Each Data_definition_indentured_list_by_part_with_document_references_to_parts has method_of_entry_-tabulation defined by exactly one Indentured_list_method object. Each Indentured_list_method defines method_of_entry for zero, one, or more Data_definition_indentured_list_by_part_with_document_-references_to_parts objects.

NOTE This assertion is established through indentured_list_method.

4.3.119 Data_list to Data_list_body

Each Data_list has body defined by exactly one Data_list_body object. Each Data_list_body defines body for exactly one Data_list objects.

4.3.120 Data_list to Data_list_header

Each Data_list has list_header defined by exactly one Data_list_header object. Each Data_list_header defines list_header for exactly one Data_list objects.

4.3.121 Data_list_body to Data_list_tabulation

Each Data_list_body has data_list_tabulations defined by one or more Data_list_tabulation objects. Each Data_list_tabulation objects defines data_list_tabulations for zero, one, or more Data_list_body objects.

4.3.122 Data_list_body to Notation

Each Data_list_body has data_list_notes defined by zero, one or more Notation objects. Each Notation defines data_list_notes for zero, one, or more Data_list_body objects.

4.3.123 Data_list_body to Revision

Each Data_list_body has revision_history defined by zero, one, or more Revision objects. Each Revision object defines revision_history for zero, one, or more Data_list_body objects.

4.3.124 Data_list_entry to Contract_submission

Each Data_list_entry has available_from defined by zero, one or more Contract_submission objects. Each Contract_submission defines available_from for zero, one or more Data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.125 Data_list_entry to Header

Each Data_list_entry has list_entry defined by zero or one Header objects. Each Header defines list_entry for zero, or one, Data_list_entry objects.

NOTE This assertion is established through valid_tdp_elements_for_data_list.

4.3.126 Data_list_entry to Identifier

Each Data_list_entry has entry_item_change_level defined by zero or one Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more Data_list_entry objects.

4.3.127 Data_list_entry to Notation

Each Data_list_entry has entry_notes defined by zero, one, or more Notation objects. Each Notation defines entry_notes for zero, one, or more Data_list_entry objects.

4.3.128 Data_list_entry to Reference_document

Each Data_list_entry has list_entry defined by zero or one Reference_document objects. Each Reference_-document defines list_entry for zero, one, or more Data_list_entry objects.

NOTE This assertion is established through valid_tdp_elements_for_data_list.

Each Data_list_entry has available_from defined by zero, one or more Reference_document objects. Each Reference_document defines available_from for zero, one, or more Data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.129 Data_list_entry to Sheet

Each Data_list_entry has list_entry defined by zero or one Sheet objects. Each Sheet defines list_entry for zero, one, or more Data_list_entry objects.

NOTE This assertion is established through valid_tdp_elements_for_data_list.

4.3.130 Data_list_entry to Source_location

Each Data_list_entry has available_from defined by zero, one or more Source_location objects. Each Source_location defines available_from for zero, one, or more Data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.131 Data_list_entry to Special_condition

Each Data_list_entry has special_conditions defined by zero, one, or more Special_condition objects. Each Special_condition defines special_conditions for zero, one, or more Data_list_entry objects.

4.3.132 Data_list_header to Header

Each Data_list_header has common_header defined by exactly one Header. Each Header defines common_-header for zero or one Data_list_header objects.

4.3.133 Data_list_tabulation to Data_list_entry

Each Data_list_tabulation has data_list_entries defined by zero, one or more Data_list_entry objects. Each Data_list_entry defines data_list_entries for zero, one, or more Data_list_tabulation.

4.3.134 Data_list_tabulation to Item_identification

Each Data_list_tabulation has for_item defined by zero or one Item_identification objects. Each Item_-identification defines for_item for zero, one, or more Data_list_tabulation objects.

4.3.135 Date to An_integer

Each Date has month defined by zero or one An_integer objects. Each An_integer defines month for zero, one, or more Date objects.

Each Date has day defined by zero or one An_integer objects. Each An_integer defines day for zero, one, or more Date objects.

Each Date has week defined by zero or one An_integer objects. Each An_integer defines week for zero, one, or more Date objects.

Each Date has year defined by exactly one An_integer object. Each An_integer defines year for zero, one, or more Date objects.

4.3.136 Date to Time

Each Date has specific_time defined by zero or one Time objects. Each Time defines specific_time for zero, one, or more Date objects.

4.3.137 Date_effectivity to Date

Each Date_effectivity has start_date defined by exactly one Date. Each Date defines start_date for zero, one, or more Date_effectivity objects.

Each Date_effectivity has end_date defined by zero or one Date objects. Each Date defines end_date for zero, one, or more Date_effectivity objects.

4.3.138 Delivery_accounting to Identifier

Each Delivery_accounting has reference_identification defined by exactly one Identifier object. Each Identifier defines reference_identification for zero, one, or more Delivery_accounting objects.

4.3.139 Delivery_accounting to Text

Each Delivery_accounting has reference_identification_description defined by exactly one Text object. Each Text defines reference_identification_description for zero, one, or more Delivery_accounting objects.

4.3.140 Design_authority to Company

Each Design_authority has design_activity_identification defined by exactly one Company object. Each Company defines design_activity_identification for zero, one, or more Design_authority objects.

4.3.141 Design_authority to Company_code

Each Design_authority has design_activity_code defined by zero or one Company_code objects. Each Company_code defines design_activity_code for zero, one, or more Design_authority objects.

4.3.142 Design_authority to Label

Each Design_authority has role defined by zero or one Label object. Each Label defines role for zero, one, or more Design_authority objects.

4.3.143 Design_authority to Person

Each Design_authority has person_responsible defined by zero or one Person objects. Each Person defines person_responsible for zero, one, or more Design_authority objects.

4.3.144 Distribution_notice to Label

Each Distribution_notice has distribution_code defined by zero or one Label objects. Each Label defines distribution_code for zero, one, or more Distribution_notice objects.

NOTE There are rules related to the Distribution_notice object that requires the existence of at least one of the attributes.

4.3.145 Distribution_notice to Person_and_organization

Each Distribution_notice has distribution_authority defined by exactly one Person_and organization object. Each Person_and_organization defines distribution_authority for zero, one, or more Distribution_notice objects.

NOTE There are rules related to the Distribution_notice object that requires the existence of at least one of the attributes.

4.3.146 Distribution_notice to Text

Each Distribution_notice has distribution_statement defined by zero or one Text objects. Each Text defines distribution_statement for zero, one, or more Distribution_notice objects.

NOTE There are rules related to the Distribution_notice object that requires the existence of at least one of the attributes.

4.3.147 Document_usage_parameter to Label

Each Document_usage_parameter has subject defined by exactly one Label objecs. Each Label defines subject for zero, one, or more Document_usage_parameter objects.

4.3.148 Document_usage_parameter to Text

Each Document_usage_parameter has value_of defined by exactly one Text objecs. Each Text defines value_of for zero, one, or more Document_usage_parameter objects.

4.3.149 Drawing to Header

Each Drawing has heading defined by exactly one Header object. Each Header defines heading for zero or one Drawing objects.

4.3.150 Drawing to Sheet

Each Drawing has pages defined by one or more Sheet objects. Each Sheet defines pages for zero, one, or more Drawing objects.

4.3.151 Drawing_suffix_number_combination to Element_identification

Each drawing_suffix_number_combination has drawing_number defined by exactly one Element_-identification object. Each Element_identification defines drawing_number for zero, one, or more Drawing_-suffix_number_combination objects.

4.3.152 Drawing_suffix_number_combination to Identifier

Each Drawing_suffix_number_combination has suffix_identifier defined by exactly one Identifier object. Each Identifier defines suffix_identifier for zero, one, or more Drawing_suffix_number_combination objects.

4.3.153 Effectivity to Approval

Each Effectivity has effectivity_approvals defined by zero, one, or more Approval objects. Each Approval defines effectivity_approvals for zero, one, or more Effectivity objects.

4.3.154 Effectivity to Label

Each Effectivity has name defined by zero or one Label objects. Each Label defines name for zero, one, or more Effectivity objects.

Each Effectivity has work_activity defined by zero or one Label objects. Each Label defines work_activity for zero, one, or more Effectivity objects.

4.3.155 Effectivity to Product_configuration

Each Effectivity has product defined by exactly one Product_configuration object. Each Product_-configuration defines product for zero, one, or more Effectivity objects.

4.3.156 Effectivity to Text

Each Effectivity has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Effectivity objects.

4.3.157 Element_identification to Alternate_element_identification

Each Element_identification has alternate_identifications defined by zero, one, or more Alternate_element_-identification objects. Each Alternate_element_identification defines alternate_identifications for zero, one, or more Element_identification objects.

NOTE This assertion is established through alternate_identification_element_select.

4.3.158 Element_identification to Certification

Each Element_identification has element_certification defined by zero or one Certification objects. Each Certification defines element_certification for zero, one, or more Element_identification objects.

4.3.159 Element_identification to Change_identification

Each Element_identification has change_status defined by zero or one Change_identification objects. Each Change_identification defines change_status for zero, one, or more Element_identification objects.

4.3.160 Element_identification to Design_authority

Each Element_identification has design_activities defined by one or more Design_authority object. Each Design_authority defines design_activities for zero, one, or more Element_identification objects.

4.3.161 Element_identification to Element_type

Each Element_identification has element_classifications defined by one, or more Element_type objects. Each Element_type defines element_classifications for zero, one, or more Element_identification objects.

4.3.162 Element_identification to Identifier

Each Element_identification has alternate_identifications defined by zero, one, or more Identifier objects. Each Identifier defines alternate_identifications for zero, one, or more Element_identification objects.

NOTE This assertion is established through alternate_identification_element_select.

Each Element_identification has identifying_number defined by exactly one Identifier object. Each Identifer defines identifying_number for zero, one, or more Element_identification objects.

4.3.163 Element_identification to Other_reference_document

Each Element_identification has outstanding_changes defined by zero, one, or more Other_reference_-document objects. Each Other_reference_document defines outstanding_changes for zero, one, or more Element_identification objects.

4.3.164 Element_identification to Text

Each Element_identification has title defined by zero or one Text objects. Each Text defines title for zero, one, or more Element_identification objects.

4.3.165 Element_type to Company

Each Element_type has code_administrator defined by zero or one Company object. Each Company defines code_administrator for zero, one, or more Element_type objects.

4.3.166 Element_type to Label

Each Element_type has element_code defined by exactly one Label object. Each Label defines element_code for zero, one, or more Element_type objects.

4.3.167 Element_type to Text

Each Element_type has type_of_coding_scheme defined by zero or one Text. Each Text defines type_of_-coding_scheme for zero, one, or more Element_type objects.

4.3.168 Exchange_file to Label

Each Exchange_file has included_in_exchange defined by exactly one Label object. Each Label defines included_in_exchange for zero, one, or more Exchange_file objects.

4.3.169 Event to Date

Each Event has actual_date defined by zero or one Date objects. Each Date defines actual_date for zero, one, or more Event objects.

Each Event has planned_date defined by zero or one Date objects. Each Date defines planned_date for zero, one, or more Event objects.

4.3.170 Event to Identifier

Each Event has id defined by exactly one Identifier object. Each Identifier defines id for zero, one, or more Event objects.

4.3.171 Event to Label

Each Event has name defined by exactly one Label object. Each Label defines name for zero, one, or more Event objects.

4.3.172 Event to Person_and_organization

Each Event has responsible_person_organization defined by zero or one Person_and_organization objects. Each Person_and_organization defines responsible_person_organization for zero, one, or more Event objects.

4.3.173 Event to Project

Each Event has assignment defined by zero or one Project objects. Each Project defines assignment for zero, one, or more Event objects.

4.3.174 Event to Text

Each Event has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Event objects.

4.3.175 Exchange_file to System_declaration

Each Exchange_file has destination_system defined by zero or one System_declaration objects. Each System_declaration defines destination_system for zero, one, or more Exchange_file objects.

4.3.176 Explicit_graphics to Geometry

Each Explicit_graphics has table_geometry defined by exactly one Geometry object. Each Geometry defines table_geometry for zero, one, or more Explicit_graphic objects.

4.3.177 External_graphics_file to A_real

Each External_graphics has file_loadpoint_x defined by exactly one A_real object. Each A_real defines file_loadpoint_y for zero, one, or more External_graphic objects.

Each External_graphics has file_loadpoint_y defined by exactly one A_real object. Each A_real defines file_loadpoint_x for zero, one, or more External_graphics objects.

4.3.178 External_graphics_file to File

Each External_graphics_file has file_id defined by exactly one File object. Each File defines file_id for zero, one, or more External_graphics objects.

4.3.179 External_library_reference to Identifier

Each External_library_reference has external_id defined by exactly one Identifier object. Each Identifier defines external_id for zero, one, or more External_library_reference objects.

Each External_library_reference has library_type defined by exactly one Identifier object. Each Identifier defines library_type for zero, one, or more External_library_reference objects.

4.3.180 External_library_reference to Text

Each External_library_reference has description defined by zero or one Text objects. Each Text defines description for zero, one, or more External_library_reference objects.

4.3.181 Field_content_holder to Field_location

Each Field_content_holder has location defined by exactly one Field_location object. Each Field_location defines location for zero, one, or more Field_content_holder objects.

4.3.182 Field_content_holder to Label

Each Field_content_holder has text_style defined by exactly one Label object. Each Label defines text_style for zero, one, or more Field_content_holder objects.

4.3.183 Field_content_holder to Text

Each Field_content_holder has field_justification defined by exactly one Text object. Each Text defines field_justification for zero, one, or more Field_content_holder objects.

4.3.184 Field_content_holder_singular to Text

Each Field_content_holder_singular has field_text defined by exactly one Text object. Each Text defines field_text for zero, one, or more Field_content_holder_singular objects.

4.3.185 Field_content_holder_tabulation to Text

Each Field_content_holder_tabulation has field_text defined by one or more Text objects. Each Text defines field_text for zero, one or more Field_content_holder_tabulation objects.

4.3.186 Field_location_definition to Load_point

Each Field_location_definition has field_load_point defined by exactly one Load_point object. Each Load_-point defines field_load_point for zero, one, or more Field_location_definition objects.

4.3.187 Field_title to Field_location

Each Field_title has location defined by exactly one Field_location object. Each Field_location defines location for zero, one, or more Field_title objects.

4.3.188 Field_title to Label

Each Field_title has field_style defined by exactly one Label object. Each Label defines field_style for zero, one, or more Field_title objects.

4.3.189 Field_title to Text

Each Field_title has field_justification defined by exactly one Text object. Each Text defines field_-justification for zero, one, or more Field_title objects.

Each Field_title has field_text defined by exactly one Text object. Each Text defines field_text for zero, one, or more Field_title objects.

4.3.190 File to Change_identification

Each File has change_status defined by zero or one Change_identification objects. Each Change_-identification defines change_status for zero, one, or more File objects.

4.3.191 File to Content_property

Each File has file_content_property defined by zero or one Content_property objects. Each Content_-property defines file_content_property for zero, one, or more File objects.

4.3.192 File to Distribution_notice

Each File has distribution_authorizations defined by zero, one, or more Distribution_notice objects. Each Distribution_notice defines distribution_authorizations for zero, one, or more File objects.

4.3.193 File to File_format

Each File has context_file_format defined by zero or one File_format objects. Each File_format defines context_file_format for zero, one, or more File objects.

4.3.194 File to Identifier

Each File has context_file_name defined by zero or one Identifer objects. Each Identifier defines context_-file_name for zero, one, or more File objects.

Each File has native_format_file_name defined by zero or one Identifier objects. Each Identifier defines native_format_file_name for zero, one, or more File objects.

4.3.195 File to Label

Each File has file_content_type defined by zero or one Label objects. Each Label defines file_content_type for zero, one, or more File objects.

4.3.196 File to Notation

Each File has file_note defined by zero or one Notation objects. Each Notation defines file_note for zero, one, or more File objects.

4.3.197 File to Person_and_organization

Each File has assigned_to defined by zero, one or more Person_and_organization objects. Each Person_-and_organization defines assigned_to for zero, one, or more File objects.

4.3.198 File to Security_classification

Each File has security_identifications defined by zero, one, or more Security_classification objects. Each Security_classification defines security_identifications for zero, one, or more File objects.

4.3.199 File to Size_characteristics

Each File has size defined by zero or one Size_characteristic objects. Each Size_characteristics defines size for zero, one, or more File objects.

4.3.200 File to System_declaration

Each File has source_system defined by zero or one System_declaration objects. Each System_declaration defines source_system for zero, one, or more File objects.

4.3.201 File_format to Date

Each File_format has release_date defined by zero or one Date objects. Each Date defines release_date for zero, one, or more File_format objects.

NOTE There are rules related to the File_format object that requires the existence of at least one of the attributes.

4.3.202 File_format to Identifier

Each File_format has format_code defined by zero or one Identifier objects. Each Identifier defines format_-code for zero, one, or more File_format objects.

NOTE 1 There are rules related to the File_format object that requires the existence of at least one of the attributes.

Each File_format has revision_level defined by zero or one Identifier objects. Each Identifier defines revision_level for zero, one, or more File_format objects.

NOTE 2 There are rules related to the File_format object that requires the existence of at least one of the attributes.

Each File_format has change_level defined by zero or one Identifier objects. Each Identifier defines change_-level for zero, one, or more File_format objects.

NOTE 3 There are rules related to the File_format object that requires the existence of at least one of the attributes.

4.3.203 File_format to Text

Each File_format has format_standard defined by zero or one Text objects. Each Text defines format_-standard for zero, one, or more File_format objects.

NOTE There are rules related to the File_format object that requires the existence of at least one of the attributes.

4.3.204 File_relationship to File

Each File_relationship has related_file defined by exactly one File object. Each File defines related_file for zero, one, or more File_relationship objects.

Each File_relationship has relating_file defined by exactly one File object. Each File defines relating_file for zero, one, or more File_relationship objects.

4.3.205 File_relationship to Label

Each File_relationship has name defined by exactly one Label object. Each Label defines name for zero, one, or more File_relationship objects.

4.3.206 File_relationship to Text

Each File_relationship has description defined by zero or one Text objects. Each Text defines description for zero, one, or more File_relationship objects.

4.3.207 General_item_definition_relationship to Item

Each General_item_definition_relationship has base_item_definition defined by exactly one Item object. Each Item defines base_item_definition for zero, one, or more General_item_definition_relationship objects.

4.3.208 General_item_definition_relationship to Label

Each General_item_definition_relationship has relation_type defined by exactly one Label object. Each Label defines relation_type for zero, one, or more General_item_definition_relationship objects.

4.3.209 General_item_definition_relationship to Text

Each General_item_definition_relationship has description defined by zero or one Text objects. Each Text defines description for zero, one, or more General_item_definition_relationship objects.

4.3.210 Geometric_validation_property to A_real

Each Geometric_validation_property has property_value defined by exactly one A_real object. Each A_real defines property_value for zero, one, or more Geometric_validation_property objects.

4.3.211 Geometric_validation_property to Label

Each Geometric_validation_property has property_type defined by exactly one Label object. Each Label defines property_type for zero, one, or more Geometric_validation_property objects.

4.3.212 Group_parent_column_header to Field_title

Each Group_parent_column_header has single_group_parent_column_header_titles defined by one or more Field_title objects. Each Field_title defines single_group_parent_column_header_titles for zero, one, or more Group_parent_column_header objects.

4.3.213 Group_parent_content_holder to Field_content_holder_tabulation

Each Group_parent_content_holder has group_parent_content_presentation defined by exactly one field_-content_holder_tabulation object. Each Field_content_holder_tabulation defines group_parent_content_-presentation for zero, one, or more Group_parent_content_holder objects.

4.3.214 Group_parent_header to Field_title

Each Group_parent_header has single_group_parent_header_titles defined by one or more Field_title objects. Each Field_title defines single_group_parent_header_titles for zero, one, or more Group_parent_-header objects.

4.3.215 Header to An_integer

Each Header has sheet_count defined by zero or one An_integer objects. Each An_integer defines sheet_-count for zero, one, or more Header objects.

4.3.216 Header to Header_configuration_with_element_identification

Each Header has header_configuration defined by one Header_configuration_with_element_identification object. Each Header_configuration_with_element defines header_configuration for zero or one Header objects.

4.3.217 Header to Label

Each Header has languages defined by zero, one, or more Label objects. Each Label defines languages for zero, one, or more Header objects.

4.3.218 Header to Size_characteristics_full_size

Each Header has size_of_sheet defined by zero or one Size_characteristics_full_size objects. Each Size_-characteristics_full_size defines size_of_sheet for zero, one, or more Header objects.

4.3.219 Header to Text

Each Header has document_keywords defined by zero, one or more Text objects. Each Text defines document_keywords for zero, one, or more Header objects.

Each Header has document_abstract defined by zero or one Text objects. Each Text defines document_-abstract for zero, one, or more Header objects.

4.3.220 Header_block to Field_content_holder_singular

Each Header_block has single_block_content_presentation defined by zero, one, or more Field_content_-holder_singular objects. Each Field_content_holder_singular defines single_block_content_presentation for zero, one, or more Header_block objects.

NOTE There are rules related to the Change identification object that requires the existence of at least one of the attributes.

4.3.221 Header_block to Field_title

Each Header_block has single_block_titles defined by zero, one, or more Field_title objects. Each Field_title defines single_block_titles for zero, one, or more Header_block objects.

NOTE There are rules related to the Change identification object that requires the existence of at least one of the attributes.

4.3.222 Header_configuration_with_element_identification to Configuration

Each Header_configuration_with_element_identification has data_configuration defined by zero or one Configuration object. Each Configuration defines data_configuration for zero, one, or more Header_-configuration_with_element_identification objects.

4.3.223 Header_configuration_with_element_identification to Element_-identification

Each Header_configuration_with_element_identification has identification defined by exactly one Element_-identification object. Each Element_identification defines identification for zero, one, or more Header_-configuration objects.

4.3.224 Header_presentation to Header_block

Each Header_presentation has header_blocks_presentation defined by one or more Header_block objects. Each Header_block defines header_blocks_presentation for zero, one, or more Header_presentation objects.

4.3.225 Header_presentation to Tabulation

Each Header_presentation has header_tabulation_presentation defined by zero, one, or more Tabulation objects. Each Tabulation defines header_tabulation_presentation for zero, one, or more Header_presentation objects.

4.3.226 Indentured_data_list to Indentured_data_list_body

Each Indentured_data_list has body defined by exactly one Indentured_data_list_body object. Each Indentured_data_list_body defines body for exactly one Indentured_data_list objects.

4.3.227 Indentured_data_list to Indentured_data_list_header

Each Indentured_data_list has header defined by exactly one Indentured_data_list_header object. Each Indentured_data_list_header defines header for exactly one Indentured_data_list objects.

4.3.228 Indentured_data_list_body to Element_identification

Each Indentured_data_list_body has standardization_documents_list defined by zero, one, or more Element_identification objects. Each Element_identification defines standardization_documents_list for zero, one, or more Indentured_data_list_body objects.

4.3.229 Indentured_data_list_body to Indentured_data_list_tabulation

Each Indentured_data_list_body has indentured_tabulation defined by exactly one Indentured_data_list_-tabulation object. Each Indentured_data_list_tabulation defines indentured_tabulation for one Indentured_-data_list_body objects.

4.3.230 Indentured_data_list_body to Notation

Each Indentured_data_list_body has notes_list defined by zero, one, or more Notation objects. Each Notation defines notes_list for zero, one, or more Indentured_data_list_body objects.

4.3.231 Indentured_data_list_body to Revision

Each Indentured_data_list_body has revision_history defined by zero, one, or more Revision objects. Each Revision defines revision_history for zero, one, or more Indentured_data_list_body objects.

4.3.232 Indentured_data_list_entry to Configuration

Each Indentured_data_list_entry has entry_configuration defined by zero or one Configuration objects. Each Configuration defines entry_configuration for zero, one, or more Indentured_data_list_entry objects.

4.3.233 Indentured_data_list_entry to Content_property

Each Indentured_data_list_entry has entry_content_property defined by zero or one Content_property objects. Each Content_property defines entry_content_property for zero, one, or more Indentured_data_-list_entry objects.

4.3.234 Indentured_data_list_entry to Contract_submission

Each Indentured_data_list_entry has available_from defined by zero, one, or more Contract_submission objects. Each Contract_submission defines available_from for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.235 Indentured_data_list_entry to Effectivity

Each Indentured_data_list_entry has effective_on defined by zero or one Effectivity objects. Each Effectivity defines effective_on for zero, one, or more Indentured_data_list_entry objects.

4.3.236 Indentured_data_list_entry to Element_identification

Each Indentured_data_list_entry has superseded_entry defined by zero or one element_identification objects. Each Element_identification defines superseded_entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through identification_select.

4.3.237 Indentured_data_list_entry to Identifier

Each Indentured_data_list_entry has entry_item_change_level defined by zero or one Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more Indentured_data_list_entry objects.

4.3.238 Indentured_data_list_entry to Item_identification

Each Indentured_data_list_entry has superseded_entry defined by zero or one Item_identification objects. Each Item_identification defines superseded_entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through identification_select.

4.3.239 Indentured_data_list_entry to Item_parent_to_item_child_relationship

Each Indentured_data_list_entry has entry defined by exactly one Item_parent_to_item_child_relationship object. Each Item_parent_to_item_child_relationship defines entry for zero, one, or more Indentured_data_-list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.240 Indentured_data_list_entry to Item_parent_to_tdp_element_child_-relationship

Each Indentured_data_list_entry has entry defined by exactly one Item_parent_to_tdp_element_child_-relationship object. Each Item_parent_to_tdp_element_child_relationship defines entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.241 Indentured_data_list_entry to Label

Each Indentured_data_list_entry has indenture_level defined by zero or one Label objects. Each Label defines indenture_level for zero, one, or more Indenture_data_list_entry objects.

4.3.242 Indentured_data_list_entry to Notation

Each Indentured_data_list_entry has entry_notes defined by zero, one, or more Notation objects. Each Notation defines entry_notes for zero, one, or more Indentured_data_list_entry objects.

4.3.243 Indentured_data_list_entry to Reference_document

Each Indentured_data_list_entry has available_from defined by zero, one, or more Reference_document objects. Each Reference_document defines available_from for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.244 Indentured_data_list_entry to Retrofit_usage

Each Indentured_data_list_entry has retrofit defined by zero or one Retrofit_usage objects. Each Retrofit_-usage defines retrofit for zero, one, or more Indentured_data_list_entry objects.

4.3.245 Indentured_data_list_entry to Source_location

Each Indentured_data_list_entry has available_from defined by zero, one, or more Source_location objects. Each Source_location defines available_from for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through source_identification.

4.3.246 Indentured_data_list_entry to Special_condition

Each Indentured_data_list_entry has special_conditions defined by zero, one, or more Special_condition objects. Each Special_condition defines special_conditions for zero, one, or more Indentured_data_list_entry objects.

4.3.247 Indentured_data_list_entry to Tdp_element_parent_to_item_child_-relationship

Each Indentured_data_list_entry has entry defined by exactly one Tdp_element_parent_to_item_child_-relationship object. Each Tdp_element_parent_to_item_child_relationship defines entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.248 Indentured_data_list_entry to Tdp_element_parent_to_tdp_element_-child_relationship

Each Indentured_data_list_entry has entry defined by exactly one Tdp_element_parent_to_tdp_element_child_relationship object. Each Tdp_element_parent_to_tdp_element_child_relationship defines entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.249 Indentured_data_list_entry to Tdp_indentured_item

Each Indentured_data_list_entry has entry defined by exactly one Tdp_indentured_item object. Each Tdp_-indentured_item defines entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.250 Indentured_data_list_entry to Tdp_indentured_tdp_element

Each Indentured_data_list_entry has entry defined by exactly one Tdp_indentured_tdp_element object. Each Tdp_indentured_tdp_element defines entry for zero, one, or more Indentured_data_list_entry objects.

NOTE This assertion is established through list_entry_select.

4.3.251 Indentured_data_list_header to Element_identification

Each Indentured_data_list_header has procurement_references defined by zero, one, or more Element_-identification objects. Each Element_dentification has procurement_references defined by zero, one, or more Indentured_data_list_header objects.

NOTE This assertion is established through Identification_select.

4.3.252 Indentured_data_list_header to Header

Each Indentured_data_list_header has common_header defined by exactly one Header object. Each Header defines common_header for zero or one Indentured_data_list_header objects.

4.3.253 Indentured_data_list_header to Item_identification

Each Indentured_data_list_header has procurement_references defined by zero, one, or more Item_-identification objects. Each Item_dentification has procurement_references defined by zero, one, or more Indentured_data_list_header objects.

NOTE This assertion is established through Identification_select.

4.3.254 Indentured_data_list_tabulation to Indentured_data_list_entry

Each Indentured_data_list_tabulation has tabulation_of_entries defined by one or more Indentured_data_-list_entry objects. Each Indentured_data_list_entry defines tabulation_of _entries for zero, one, or more Indentured_data_list_tabulation objects.

4.3.255 Indentured_data_list_tabulation to Indentured_list_by_document

Each Indentured_data_list_tabulation has method_of_entry_tabulation defined by one Indentured_list_by_-document object. Each Indentured_list_by_document defines method_of_entry_tabulation for zero, one, or more Indentured_data_list_tabulation objects.

NOTE This assertion is established through indentured_list_method.

4.3.256 Indentured_data_list_tabulation to Indentured_list_by_part

Each Indentured_data_list_tabulation has method_of_entry_tabulation defined by exactly one Indentured_-list_by_part object. Each Indentured_list_by_part defines method_of_entry_tabulation for zero, one, or more Indentured_data_list_tabulation objects.

NOTE This assertion is established through indentured_list_method.

4.3.257 Indentured_data_list_tabulation to Indentured_list_by_part_with_-document_references_to_parts

Each Indentured_data_list_tabulation has method_of_entry_tabulation defined by exactly one Indentured_-list_by_part_with_document_references_to_part object. Each Indentured_list_by_part_with_document_-references_to_parts defines method_of_entry_tabulation for zero, one, or more Indentured_data_list_-tabulation objects.

NOTE This assertion is established through indentured_list_method.

4.3.258 Indentured_list_by_document to Tdp_indentured_tdp_element

Each Indentured_list_by_document has top_indentured_tdp_elements defined by one or more Tdp_-indentured_tdp_element objects. Each Tdp_indentured_tdp_element object defines top_indentured_tdp_-element for zero, one, or more Indentured_list_by_document objects.

4.3.259 Indentured_list_by_part to Tdp_indentured_item

Each Indentured_list_by_part has top_indentured_items defined by one or more Tdp_indentured_item objects. Each Tdp_indentured_item defines top_indentured_items for zero, one, or more Indentured_list_-by_part objects.

4.3.260 Indentured_list_by_part_with_document_references_to_parts to Tdp_-indentured_item

Each Indentured_list_by_part_with_document_reference_to_parts has top_indentured defined by zero, one or more Tdp_indentured_item objects. Each Tdp_indentured_item defines top_indentured_ for zero, one, or more Indentured_list_by_part_with_document_reference_to_parts objects.

NOTE This assertion is established through item_or_element.

4.3.261 Indentured_list_by_part_with_document_references_to_parts to Tdp_-indentured_tdp_element

Each Indentured_list_by_part_with_document_reference_to_parts has top_indentured defined by zero, one or more Tdp_indentured_tdp_element objects. Each Tdp_indentured_tdp_element defines top_indentured_ for zero, one, or more Indentured_list_by_part_with_document_reference_to_parts objects.

NOTE This assertion is established through item_or_element.

4.3.262 Independent_property to External_library_reference

Each Independent_property has property_source defined by zero or one External_library_reference objects. Each External_library_reference defines property_source for zero, one, or more Independent_property objects.

4.3.263 Independent_property to Identifier

Each Independent_property has id defined by exactly one Identifier object. Each Identifier defines id for zero, one, or more Independent_property objects.

4.3.264 Independent_property to Measure

Each Independent_property has allowed_units defined by zero, one, or more Measure objects. Each Measure defines allowed_units for zero, one, or more Independent_property objects.

4.3.265 Independent_property to Text

Each Independent_property has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Independent_property objects.

4.3.266 Independent_property_relationship to Independent_property

Each Independent_property_relationship has related_property defined by exactly one Independent_property object. Each Independent_property defines related_property for zero, one, or more Independent_property_-relationship objects.

Each Independent_property_relationship has relating_property defined by exactly one Independent_property object. Each Independent_property defines relating_property for zero, one, or more Independent_property_-relationship objects.

4.3.267 Independent_property_relationship to Label

Each Independent_property_relationship has relation_type defined by exactly one Label object. Each Label defines relation_type for zero, one, or more Independent_property_relationship objects.

4.3.268 Independent_property_relationship to Text

Each Independent_property_relationship has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Independent_property_relationship objects.

4.3.269 Independent_property_usage to Independent_property

Each Independent_property_usage has assigned_property defined by exactly one Independent_property object. Each Independent_property defines assigned_property for zero, one, or more Independent_property_-usage objects.

4.3.270 Independent_property_usage to Property_assignment

Each Independent_property_usage has assigned_to defined by exactly one Property_assignment object. Each Property_assignment defines assigned_to for zero, one, or more Independent_property_usage objects.

4.3.271 Index_list to Index_list_body

Each Index_list has body defined by exactly one Index_list_body object. Each Index_list_body defines body for one Index_list objects.

4.3.272 Index_list to Index_list_header

Each Index_list has list_header defined by exactly one Index_list_header object. Each Index_list_header defines list_header for exactly one Index_list object.

4.3.273 Index_list_body to Index_list_entry

Each Index_list_body has index_list_entries defined by one or more Index_list_entry objects. Each Index_-list_entry defines index_list_entries for zero, one, or more Index_list_body objects.

4.3.274 Index_list_body to Notation

Each Index_list_body has index_list_notes defined by zero, one, or more Notation objects. Each Notation defines index_list_notes for zero, one, or more Index_list_body objects.

4.3.275 Index_list_body to Revision

Each Index_list_body has revision_history defined by zero, one, or more Revision objects. Each Revision defines revision_history for zero, one, or more Index_list_body objects.

4.3.276 Index_list_entry to Header

Each Index_list_entry has list_entry defined by zero or one Header objects. Each Header defines list_entry for zero or one, Index_list_entry objects.

4.3.277 Index_list_entry to Identifier

Each Index_list_entry has entry_item_change_level defined by zero or one Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more Index_list_entry objects.

4.3.278 Index_list_entry to Notation

Each Index_list_entry has entry_notes defined by zero, one or more Notation objects. Each Notation defines entry_notes for zero, one, or more Index_list_entry objects.

4.3.279 Index_list_header to Header

Each Index_list_header has common_header defined by exactly one Header object. Each Header defines common_header for zero or one Index_list_header objects.

4.3.280 Item to Alternate_item

Each Item has alternates define by zero, one, or more Alternate_item objects. Each Alternate_item defines alternates for zero, one, or more Item objects.

4.3.281 Item to Configuration

Each Item has item_configuration defined by zero or one Configuration objects. Each Configuration defines item_configuration for zero, one, or more Item objects.

4.3.282 Item to File

Each Item has shape defined by zero or one File objects. Each File defines shape for zero, one, or more Item objects.

Each Item has shape_aspects defined by zero, one, or more File objects. Each File defines shape_aspect for zero, one, or more Item objects.

4.3.283 Item to Item_identification

Each Item has identification defined by exactly one Item_identification object. Each Item_identification defines identification for zero, one, or more Item objects.

4.3.284 Item to Label

Each Item has item_contexts defined by zero, one, or more Label objects. Each Label defines item_contexts for zero, one, or more Item objects.

4.3.285 Item to Notation

Each Item has notes defined by zero, one, or more Notation objects. Each Notation defines notes for zero, one, or more Item objects.

4.3.286 Item to Reference_document

Each Item has conditions_defined_through_simple_reference defined by zero, one, or more Reference_-document_objects. Each Reference_document defines conditions_defined_through_simple_reference for zero, one, or more Item objects.

4.3.287 Item to Reference_document_usage

Each Item has conditions_defined_through_constrained_document defined by zero, one, or more Reference_-document_usage objects. Each Reference_document_usage defines conditions_defined_through_-constrained_document for zero, one, or more Item objects.

4.3.288 Item to Shape_model

Each Item has shape defined by zero or one Shape_model objects. Each Shape_model defines shape for zero, one, or more Item objects.

Each Item has shape_aspects defined by zero, one, or more Shape_model objects. Each Shape_model defines shape_aspect for zero, one, or more Item objects.

4.3.289 Item to Special_condition

Each Item has special_conditions defined by zero, one or more Special_condition objects. Each Special_-condition defines special_conditions for zero, one, or more Item objects.

4.3.290 Item to Weight

Each Item has item_weight defined by zero or one Weight objects. Each Weight defines item_weight for zero, one, or more Item objects.

4.3.291 Item_identification to Alternate_identification_of_item

Each Item_identification has alternate_identifications defined by zero, one or more Alternate_identification_-of_item objects. Each Alternate_identification_of_item defines alternate_identifications for zero, one, or more Item_identification objects.

NOTE This assertion is established through alterante_identification_item_select.

4.3.292 Item_identification to Certification

Each Item_identification has item_certification defined by zero or one Certification objects. Each Certification defines item_certification for zero, one, or more Item_identification objects.

4.3.293 Item_identification to Change_identification

Each Item_identification has change_status defined by zero or one Change_identification objects. Each Change_identification defines change_status for zero, one, or more Item_identification objects.

4.3.294 Item_identification to Design_authority

Each Item_identification has design_activity defined by exactly one Design_authority object. Each Design_-authority defines design_activity for zero, one, or many Item_identification object.

4.3.295 Item_identification to Drawing_suffix_number_combination

Each Item_identification has identifying_number defined by zero or one Drawing_suffix_number_-combination objects. Each Drawing_suffix_number_combination defines identifying_number for zero, one, or more Item_identification objects.

NOTE This assertion is established through item_identifying_number_select.

4.3.296 Item_identification to Identifier

Each Item_identification has alternate_identifications defined by zero, one or more Identifier objects. Each Identifier defines alternate_identifications for zero, one, or more Item_identification objects.

NOTE This assertion is established through alterante_identification_item_select.

Each Item_identification has identifying_number defined by zero or one Identifier objects. Each Identifier defines identifying_number for zero, one, or more Item_identification objects.

NOTE This assertion is established through item_identifying_number_select.

4.3.297 Item_identification to Item_type

Each Item_identification has classifications defined by one or more Item_type objects. Each Item_type defines classifications for zero, one, or more Item_identification objects.

4.3.298 Item_identification to Label

Each Item_identification has classifications defined by zero, one, or more Label objects. Each Label defines classifications for zero, one, or more Item_identification objects.

Each Item_identification has nomenclature_or_name defined by zero or one Label objects. Each Label defines nomenclature_or_name for zero, one, or more Item_identification objects.

4.3.299 Item_identification to Source_information_type

Each Item_identification has source_information defined by zero or one Source_information_type objects. Each Source_information_type defines nomenclature_or_name for zero, one, or more Item_identification objects.

4.3.300 Item_list to Item_identification

Each Item_list has list_for defined by zero or one Item_identification objects. Each Item_identification defines list_for for zero, one, or more Item_list objects.

NOTE This assertion is established through item_identifying_number_select.

4.3.301 Item_list to Item_list_drawing

Each Item_list has list_for defined by zero or one Item_list_drawing objects. Each Item_list_drawing defines list_for for zero, one, or more Item_list objects.

NOTE This assertion is established through item_list_applies_to_select.

4.3.302 Item_list to Item_usage

Each Item_list has component_of defined by one or more Item_usage objects. Each Item_usage defines component_of for one Item_list objects.

4.3.303 Item_list to Text

Each Item_list has type_of_component_list defined by exactly one Text object. Each Text defines type_of_-component_list for zero, one, or more Item_list object.

NOTE This assertion is established through component_list_type_select.

4.3.304 Item_list_drawing to Element_identification

Each Item_list_drawing has drawing_number defined by exactly one Element_identification object. Each Element_identification defines drawing_number for zero, one, or more Item_list_drawing objects.

4.3.305 Item_list_drawing to Item_identification

Each Item_list_drawing has assembly_item_list_applies_to defined by zero or one Item_identification objects. Each Item_identification defines assembly_item_list_applies_to for zero, one, or more Item_list_-drawing objects.

4.3.306 Item_parent_to_item_child_relationship to Item_identification

Each Item_parent_to_item_child_relationship has parent defined by exactly one Item_identification object. Each Item_identification defines parent for zero, one, or more Item_parent_to_item_child_relationship objects.

Each Item_parent_to_item_child_relationship has child defined by exactly one Item_identification object. Each Item_identification defines child for zero, one, or more Item_parent_to_item_child_relationship objects.

4.3.307 Item_parent_to_tdp_element_child_relationship to Element_-identification

Each Item_parent_to_tdp_element_child_relationship has child defined by exactly one Element_-identification object. Each Element_identification defines child for zero, one, or more Item_parent_to_tdp_-element_child_relationship objects.

4.3.308 Item_parent_to_tdp_element_child_relationship to Item_identification

Each Item_parent_to_tdp_element_child_relationship has parent defined by exactly one Item_identification object. Each Item_identification defines parent for zero, one, or more Item_parent_to_tdp_element_child_-relationship objects.

4.3.309 Item_type to Company

Each Item_type has code_administrator defined by zero or one Company objects. Each Company defines code_administrator for zero, one, or more Item_type objects.

4.3.310 Item_type to Label

Each Item_type has type_code defined by exactly one Label object. Each Label defines type_code for zero, one, or more Item_type objects.

4.3.311 Item_type to Text

Each Item_type has type_of_coding_scheme defined by zero or one Text objects. Each Text defines type_-of_coding_scheme for zero, one, or more Item_type objects.

4.3.312 Item_usage to Effectivity

Each Item_usage has effective_on defined by zero or one Effectivity objects. Each Effectivity defines effective_on for zero, one, or more Item_usage objects.

4.3.313 Item_usage to Item

Each Item_usage has being_defined_for defined by exactly one Item object. Each Item defines being_-defined_for for zero, one, or more Item_usage objects.

4.3.314 Item_usage to List_item_usage

Each Item has list_item defined by zero or one List_item_usage objects. Each List_item_usage defines list_-item for one Item_usage objects.

4.3.315 Item_usage to Quantity

Each Item_usage has quantity_used_in_next_higher_assembly defined by zero or one Quantity objects. Each Quantity defines quantity_used_in_next_higher_assembly for zero, one, or more Item_usage objects.

4.3.316 Item_usage to Retrofit_usage

Each Item_usage has retrofit defined by zero or one Retrofit_usage objects. Each Retrofit_usage defines retrofit for zero, one, or more Item_usage objects.

4.3.317 Item_usage to Special_condition

Each Item_usage has special_conditions defined by zero, one or more Special_condition objects. Each Special_condition defines special_conditions for zero, one or more Item_usage objects.

4.3.318 List_item_usage to Approval

Each List_item_usage has approvals defined by zero, one, or more Approval objects. Each Approval defines approvals for by zero, one, or more List_item_usage objects.

4.3.319 List_item_usage to Identifier

Each List_item_usage has reference_designators defined by zero, one, or more Identifier objects. Each Identifier defines reference_designators for by zero, one, or more List_item_usage objects.

Each List_item_usage has entry_item_change_level defined by zero or one Identifier objects. Each Identifier defines entry_item_change_level for zero, one, or more List_item_usage objects.

Each List_item_usage has drawing_zones defined by zero, one, or more Identifier objects. Each Identifier defines drawing_zones for zero, one, or more List_item_usage objects.

Each List_item_usage has find_number defined by zero or one Identifier objects. Each Identifier defines find_number for zero, one, or more List_item_usage objects.

4.3.320 List_item_usage to Label

Each List_item_usage has item_type defined by zero or one Label objects. Each Label defines item_type for zero, one, or more List_item_usage objects.

4.3.321 List_item_usage to Notation

Each List_item_usage has entry_notes defined by zero, one or more Notation objects. Each Notation defines entry_notes defined by zero, one, or more List_item_usage objects.

4.3.322 List_item_usage to Security_classification

Each List_item_usage has security_information defined by zero or one Security_classification objects. Each Security_classification defines security_information for zero, one, or more List_item_usage objects.

4.3.323 List_presentation to Body_presentation

Each List_presentation has body_page1_format defined by one Body_presentation object. Each Body_-presentation defines body_page1_format for zero, one, or more List_presentation objects.

Each List_presentation has body_pagen_format defined by zero, one, or more Body_presentation objects. Each Body_presentation defines body_pagen_format for zero, one, or more List_presentation objects.

4.3.324 List_presentation to Header_presentation

Each List_presentation has header_page1_format defined by exactly one Header_presentation object. Each Header_presentation defines header_pagen_format for zero, one, or more List_presentation objects.

Each List_presentation has header_pagen_format defined by zero, one, or more Header_presentation objects. Each Header_presentation defines header_pagen_format for zero, one, or more List_presentation objects.

4.3.325 List_presentation to Measure

Each List_presentation has unit_of_length defined by exactly one Measure object. Each Measure defines unit_of_length for zero, one, or more List_presentation objects.

4.3.326 List_presentation to Page_presentation

Each List_presentation has page_format defined by exactly one Page_presentation object. Each Page_-presentation defines page_format for zero, one, or more List_presentation objects.

4.3.327 Load_point to A_real

Each Load_point has load_point_x defined by exactly one A_real object. Each A_real defines load_point_x for zero, one, or more Load_point objects.

Each Load_point has load_point_y defined by exactly one A_real object. Each A_real defines load_point_y for zero, one, or more Load_point objects.

4.3.328 Location_instance to A_real

Each Location_instance has scale defined by zero or one A_real objects. Each A_real defines scale for zero, one, or more Location_instances objects.

4.3.329 Location_instance to File

Each Location_instance has geometry_definition defined by exactly one Source_file object. Each Source_file defines geometry_definition for zero, one, or more Location_instances objects.

4.3.330 Location_instance to Geometry

Each Location_instance has geometry_definition defined by exactly one Geometry object. Each Geometry defines geometry_definition for zero, one, or more Location_instances objects.

4.3.331 Location_instance to Identifier

Each Location_instance has id defined by exactly one Identifier object. Each Identifier defines id for zero, one, or more Location_instances objects.

4.3.332 Location_instance to Transformation

Each Location_instance has transformation_definition defined by exactly one Transformation object. Each Transformation defines transformation_definition for zero, one, or more Location_instances objects.

4.3.333 Lot_effectivity to A_number

Each Lot_effectivity has lot_size defined by exactly one A_number object. Each A_number defines lot_size for zero, one, or more Lot_effectivity objects.

4.3.334 Lot_effectivity to Identifier

Each Lot_effectivity has lot_identification defined by exactly one Identifier object. Each Identifier defines lot_identification for zero, one, or more Lot_effectivity objects.

4.3.335 Lot_effectivity to Measure

Each Lot_effectivity has lot_unit_of_measure defined by exactly one Measure object. Each Measure defines lot_unit_of_measure for zero, one, or more Lot_effectivity objects.

4.3.336 Made_from_stock_material to A_number

Each Made_from_stock_material has quantity_of_item_made_from_stock defined by zero or one A_number objects. Each A_number object defines quantity_of_item_made_from_stock for zero, one, or more Made_-from_stock_material objects.

4.3.337 Made_from_stock_material to Reference_document_usage

Each Made_from_stock_material has specified_by defined by zero, one, or more Reference_document_-usage objects. Each Reference_document_usage defines specified_by for zero, one, or more Made_from_-stock_material objects.

4.3.338 Made_from_stock_material to Stock_material

Each Made_from_stock_material has stock defined by zero or one Stock_material objects. Each Stock_-material defines stock for zero, one, or more Made_from_stock_material objects.

4.3.339 Made_from_stock_material to Stock_size

Each Made_from_stock_material has size defined by zero or one Stock_size objects. Each Stock_size defines size for zero, one, or more Made_from_stock_material object.

4.3.340 Make_from_relationship to Item

Each make_from_relationship has item_that_is_acting_as_stock defined by exactly one Item object. Each Item defines item_that_is_acting_as_stock for zero, one, or more Make_from_relationship objects.

4.3.341 Make_from_relationship to Number_with_unit

Each Make_from_relationship has quantity_of_item_made_from_other_item defined by exactly one Number_with_unit object. Each Number_with_unit defines quantity_of_item_made_from_other_item for zero, one, or more Make_from_relationship objects.

4.3.342 Notation to Element_identification

Each Notation has referenced_document_in_note defined by zero, one, or many Element_identification objects. Each Element_identification defines reference_code for zero, one, or more Notation objects.

4.3.343 Notation to Identifier

Each Notation has reference_code defined by zero or one Identifier objects. Each Identifier defines reference_code for zero, one, or more Notation objects.

4.3.344 Notation to Label

Each Notation has type_of_notation defined by zero or one Label objects. Each Label defines type_of_-notation for zero, one, or more Notation objects.

Each Notation has note_title defined by zero or one Label object. Each Label defines note_title for zero, one or more Notation objects.

4.3.345 Notation to Text

Each Notation has note defined by exactly one Text object. Each Text defines note for zero, one, or more Notation objects.

Each Notation has note_parameter defined by zero, one, or more Text object. Each Text defines note_-parameter for zero, one, or more Notation objects.

4.3.346 Number_with_units to A_real

Each Number_with_units has value_of defined by exactly one A_real object. Each A_real defines value_of for zero, one, or more Number_with_units objects.

4.3.347 Number_with_units to Measure

Each Number_with_units has units_of defined by exactly one Measure object. Each Measure defines units_-of for zero, one, or more Number_with_units object.

4.3.348 Other_list to Other_list_body

Each Other_list has list_body defined by exactly one Other_list_body object. Each Other_list_body defines list_body for exactly one Other_list objects.

4.3.349 Other_list to Other_list_header

Each Other_list has list_header defined by exactly one Other_list_header object. Each Other_list_header defines list_header for exactly one Other_list object.

4.3.350 Other_list_body to Notation

Each Other_list_body has notes defined by zero, one, or more Notation objects. Each Notation defines notes for zero, one, or more Other_list_body objects.

NOTE There are rules related to the Other_list_body object that requires the existence of at least one of the attributes.

4.3.351 Other_list_body to Other_list_body_table

Each Other_list_body has list_body_tables defined by zero, one, or more Other_list_body_table objects. Each Other_list_body_table defines list_body_tables for zero, one, or more Other_list_body objects.

NOTE There are rules related to the Other_list_body object that requires the existence of at least one of the attributes.

4.3.352 Other_list_body to Revision

Each Other_list_body has revision_history defined by zero, one, or more Revision objects. Each Revision defines revision_history for zero, one, or more Other_list_body objects.

4.3.353 Other_list_body_table to Label

Each Other_list_body_table has tabulation_column_descriptions defined by zero, one, or more Label objects. Each Label defines tabulation_column_descriptions for zero, one, or more Other_list_body_table objects.

Each Other_list_body has description_of_tabulation defined by zero or one Label objects. Each Label defines description_of_tabulation for zero, one, or more Other_list_body objects.

4.3.354 Other_list_body_table to Tabulation_entry

Each Other_list_body_table has tabulation_rows defined by one or more Tabulation_entry objects. Each Tabulation_entry defines tabulation_rows for zero, one, or more Other_list_body_table objects.

4.3.355 Other_list_header to Header

Each Other_list_header has common_header defined by exactly one Header object. Each Header defines common_header for zero or one Other_list_header objects.

4.3.356 Page_parameters to A_real

Each Page_parameters has page_anchor_x defined by exactly one A_real object. Each A_real defines page_-anchor_x for zero, one, or more Page_parameters objects.

Each Page_parameters has page_anchor_y defined by exactly one A_real object. Each A_real defines page_-anchor_y for zero, one, or more Page_parameters objects.

4.3.357 Page_parameters to Explicit_graphics

Each Page_parameters has graphics_declaration defined by zero or one Explicit_graphics objects. Each Explicit_graphics defines graphics_declaration for zero, one, or more Page_parameters objects.

NOTE This assertion is established through graphics_prarameters.

4.3.358 Page_parameters to External_graphics_file

Each Page_parameters has graphics_declaration defined by zero or one External_graphics_file objects. Each External_graphics_file defines graphics_declaration for zero, one, or more Page_parameters objects.

NOTE This assertion is established through graphics_prarameters.

4.3.359 Page_parameters to Text

Each Page_parameters has page_size_code defined by zero or one Text objects. Each Text defines page_-size_code for zero, one, or more Page_parameters objects.

Each Page_parameters has paper_size defined by zero or one Text objects. Each Text defines paper_size for zero, one, or more Page_parameters objects.

4.3.360 Page_presentation to Page_parameters

Each Page_presentation has page_master_parameters defined by exactly one Page_parameters object. Each Page_parameters defines page_master_parameters for zero, one, or more Page_presentation objects.

Each Page_presentation has page1_parameters defined by zero or one Page_parameters objects. Each Page_parameters defines page1_parameters for zero, one, or more Page_presentation objects.

Each Page_presentation has pagen_parameters defined by zero, one, or more Page_parameters objects. Each Page_parameters defines pagen_parameters for zero, one, or more Page_presentation objects.

4.3.361 Part_occurrence_in_assembly to Location_instance

Each Part_occurrence_in_assembly has location defined by zero or one Location_instance objects. Each Location_instance defines location for zero, one, or more Part_occurrence_in_assembly objects.

4.3.362 Parts_list to Parts_list_body

Each Parts_list has list_body defined by exactly one Parts_list_body object. Each Parts_list_body defines list_body for exactly one Parts_list object.

4.3.363 Parts_list to Parts_list_header

Each Parts_list has list_header defined by exactly one Parts_list_header object. Each Parts_list_header defines list_header for exactly one Parts_list objects.

4.3.364 Parts_list_body to Item_list

Each Parts_list_body has parts_tabulations defined by one or more Item_list objects. Each Item_list defines parts_tabulations for zero, one, or more Parts_list_body objects.

4.3.365 Parts_list_body to Notation

Each Parts_list_body has list_notes defined by zero, one, or more Notation objects. Each Notation defines list_notes for zero, one, or more Parts_list_body objects.

4.3.366 Parts_list_body to Revision

Each Parts_list_body has revision_history defined by zero, one, or more Revision objects. Each Revision defines revision_history for zero, one, or more Parts_list_body objects.

4.3.367 Parts_list_body to Tdp_element_list_item

Each Parts_list_body has referenced_tdp_elements defined by zero, one, or more Tdp_element_list_item objects. Each Tdp_element_list_item defines referenced_tdp_elements for zero, one, or more Parts_list_body objects.

4.3.368 Parts_list_header to Header

Each Parts_list_header has common_header defined by exactly one Header object. Each Header defines common_header for zero or one Parts_list_header objects.

4.3.369 Person to Identifier

Each Person has first_name defined by zero or one Identifier objects. Each Identifier defines first_name for zero, one, or more Person objects.

Each Person has middle_name defined by zero or one Identifier objects. Each Identifier defines middle_name for zero, one, or more Person objects.

Each Person has last_name defined by zero or one Identifier objects. Each Identifier defines last_name for zero, one, or more Person objects.

Each Person has unique_identifier defined by exactly one Identifier object. Each Identifier defines unique_-identifier for zero, one, or more Person objects.

NOTE There are rules related to the person_and_organization object that require the existence of at least one of the attributes.

4.3.370 Person to Text

Each Person has title defined by zero or one Text objects. Each Text defines title for zero, one, or more Person objects.

NOTE There are rules related to the person_and_organization object that require the existence of at least one of the attributes.

4.3.371 Person_and_organization to Company

Each Person_and_organization has organization defined by zero or one Company objects. Each Company defines organization for zero, one, or more Person_and_organization objects.

NOTE There are rules related to the person_and_organization object that require the existence of at least one of the attributes.

4.3.372 Person_and_organization to Person

Each Person_and_organization has person_identification defined by zero or one Person objects. Each Person defines person_identification for by zero, one, or more Person_and_organization objects.

NOTE There are rules related to the person_and_organization object that require the existence of at least one of the attributes.

4.3.373 Person_and_organization to Text

Each Person_and_organization has electronic_mail_address defined by zero or one Text objects. Each Text defines electronic_mail_address for zero, one, or more Person_and_organization objects.

Each Person_and_organization has person_address defined by zero or one Text objects. Each Text defines person_address for zero, one, or more Person_and_organization objects.

4.3.374 Product_configuration to Approval

Each Product_configuration has approval_of_configuration defined by zero, one, or more Approval objects. Each Approval defines approval_of_configuration for zero, one, or more Product_configuration objects.

4.3.375 Product_configuration to Identifier

Each Product_configuration has configuration_id defined by exactly one Identifier object. Each Identifier defines configuration_id for zero, one, or more Product_configuration objects.

4.3.376 Product_configuration to Item_identification

Each Product_configuration has design_item defined by zero or one Item_identification objects. Each Item_-identification defines design_item for zero, one, or more Product_configuration objects.

4.3.377 Product_configuration to Person_and_organization

Each Product_configuration has assigned_to defined by zero, one or more Person_and_organization objects. Each Person_and_organization defines assigned_to for zero, one, or more Product_configuration objects.

4.3.378 Product_configuration to Product_model

Each Product_configuration has product_name defined by exactly one Product_model object. Each Product_model defines product_name for zero, one, or more Product_configuration objects.

4.3.379 Product_data_set to Drawing

Each Product_data_set has related defined by zero, one, or more Drawing objects. Each Drawing defines related for zero, one, or more Product_data_set objects.

NOTE This assertion is established through drawing_or_product_data_set.

4.3.380 Product_data_set to Product_data_set

Each Product_data_set has related defined by zero, one, or more Product_data_set objects. Each Product_-data_set defines related for zero, one, or more Product_data_set objects.

NOTE This assertion is established through drawing_or_product_data_set.

4.3.381 Product_data_set to Source_file

Each Product_data_set has file_configuration defined by zero, one, or more Source_file objects. Each Source_file defines file_configuration for zero, one, or more Product_data_set objects.

4.3.382 Product_data_set_with_format to Header

Each Product_data_set_with_format has product_header defined by exactly one Header object. Each Header defines product_header for zero or one Product_data_set_with_format objects.

4.3.383 Product_data_set_with_format to Product_presentation

Each Product_data_set_with_format has presentation_of_product_data_set defined by zero, one, or more Product_presentation objects. Each Product_presentation defines presentation_of_product_data_set for zero, one, or more Product_data_set_with_format objects.

4.3.384 Product_data_set_with_shading to Header_configuration_with_-element_identification

Each Product_data_set_with_shading has data_configuration defined by exactly one Header_configuration_-with_element_identification object. Each Header_configuration_with_element_identification defines data_-configuration for zero or one Product_data_set_with_shading objects.

4.3.385 Product_data_set_with_shading to Shaded_shape_model

Each Product_data_set_with_shading has a_shaded_model defined by exactly one Shaded_shape_model object. Each Shaded_shape_model defines a_shaded_shape_model for zero, one, or more Product_data_-set_with_shading objects.

4.3.386 Product_data_set_without_format to Header_configuration_with_-element_identification

Each Product_data_set_without_format has data_configuration defined by exactly one Header_-configuration_with_element_identification object. Each Header_configuration_with_element_identification defines data_configuration for zero or one Product_data_set_without_format objects.

4.3.387 Product_data_set_without_format to Shape_model

Each Product_data_set_without_format has a_model defined by zero or one Shape_model objects. Each Shape_model defines a_model for zero, one, or more Product_data_set_without_format objects.

4.3.388 Product_model to Label

Each Product_model has model_name defined by exactly one Label object. Each Label defines model_name for zero, one, or more Product_model objects.

4.3.389 Product_model to Project

Each Product_model has projects defined by zero, one or more Project objects. Each Project defines projects for zero, one, or more Product_model objects.

4.3.390 Product_presentation to Annotation

Each Product_presentation has annotation_presentation defined by exactly one Annotation object. Each Annotation defines annotation_presentation for zero, one, or more Product_presentation objects.

4.3.391 Product_presentation to Shape_model

Each Product_presentation has part_shape_views defined by zero, one, or more Shape_model objects. Each Shape_model defines part_shape_views for zero, one, or more Product_presentation objects.

4.3.392 Project to Date

Each Project has end_date defined by zero or one Date objects. Each Date defines end_date for zero, one, or more Project objects.

Each Project has start_date defined by zero or one Date objects. Each Date defines start_date for zero, one, or more Project objects.

4.3.393 Project to Identifier

Each Project has id defined by exactly one Identifier object. Each Identifier defines id for zero, one, or more Project objects.

4.3.394 Project to Label

Each Project has name defined by exactly one Label object. Each Label defines id for zero, one, or more Project objects.

4.3.395 Project to Person_and_organization

Each Project has participants defined by one or more Person_and_organization objects. Each Person_and_-organization defines participants for zero, one, or more Project objects.

Each Project has project_owner defined by zero or one Person_and_organization objects. Each Person_-and_organization defines project_owner for zero, one, or more Project objects.

4.3.396 Project to Text

Each Project has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Project objects.

4.3.397 Property_assignment to Identifier

Each Property_assignment has described_element defined by exactly one Identifier object. Each Identifier defines described_element for zero, one, or more Property_assignment objects.

4.3.398 Property_assignment to Label

Each Property_assignment has name defined by zero or one Label objects. Each Label defines name for zero, one, or more Property_assignment objects.

4.3.399 Property_assignment to Text

Each Property_assignment has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Property_assignment objects.

4.3.400 Quantity to A_number

Each Quantity has value_of defined by zero or one A_number objects. Each A_number defines value_of for zero, one, or more Quantity objects.

NOTE This assertion is established through quantity_type_select.

4.3.401 Quantity to Text

Each Quantity has quantity_accuracy defined by zero or one Text objects. Each Text defines quantiy_-accuracy for zero, one, or more Quantity objects.

NOTE This assertion is established through quantity_accuracy_select.

Each Quantity has value_of defined by zero or one Text objects. Each Text defines value_of for zero, one, or more Quantity objects.

NOTE This assertion is established through quantity_type_select.

4.3.402 Quantity to Volume

Each Quantity has value_of defined by zero or one Volume objects. Each Volume defines value_of for zero, one, or more Item_usage objects.

NOTE This assertion is established through quantity_type_select.

4.3.403 Quantity to Weight

Each Quantity has value_of defined by zero or one Weight objects. Each Weight defines value_of for zero, one, or more Quantity objects.

NOTE This assertion is established through quantity_type_select.

4.3.404 Reason to Item_identification

Each Reason has applicable_to defined by zero, one, or more Item_identification objects. Each Item_-identification defines applicable_to for zero, one, or more Reason objects.

4.3.405 Reason to Text

Each Reason has exchange_purpose defined by zero or one Text objects. Each Text defines exchange_-purpose for zero, one, or more Reason objects.

Each Reason has base defined by zero or one Text objects. Each Text defines base for zero, one, or more Reason objects.

NOTE There exist rules related to the reason object that require the existence of at least one of the attributes.

4.3.406 Reference_document to Element_identification

Each Reference_document has document_identification defined by exactly one Element_identification object. Each Element_identification defines document_identification for zero, one, or more Reference_document objects.

4.3.407 Reference_document_usage to Document_usage_parameter

Each Reference_document_usage has usage_parameter defined by zero or one Document_usage_parameter objects. Each Document_usage_parameter defines usage_parameter for zero, one, or more Reference_-document_usage objects

4.3.408 Reference_document_usage to Identifier

Each Reference_document_usage has usage_code defined by zero or one Identifier objects. Each Identifier defines usage_code for zero, one, or more Reference_document_usage objects.

4.3.409 Reference_document_usage to Label

Each Reference_document_usage has location_in_document defined by zero or one Label objects. Each Label defines location_in_document for zero, one, or more Reference_document_usage objects.

4.3.410 Reference_document_usage to Reference_document

Each Reference_document_usage has document_referenced defined by exactly one Reference_document object. Each Reference_document defines document_referenced for zero, one, or more Reference_-document_usage objects.

4.3.411 Reference_document_usage to Text

Each Reference_document_usage has usage_description defined by zero or one Text objects. Each Text defines usage_description for zero, one, or more Reference_document_usage objects.

4.3.412 Relative_event to Event

Each Relative_event has related_event defined by exactly one Event object. Each Event defines related_event for zero, one, or more Relative_event objects.

4.3.413 Relative_event to Number_with_units

Each Relative_event has offset defined by exactly one Number_with_units object. Each Number_with_units defines offset for zero, one, or more Relative_event objects.

4.3.414 Release_authentication to Company

Each Release_authentication has release_authority defined by exactly one Company object. Each Company defines release_authority for zero, one, or more Release_authentication objects.

4.3.415 Release_authentication to Company_code

Each Release_authentication has release_authority_code defined by zero or one Company_code objects. Each Company_code defines release_authority_code for zero, one, or more Release_authentication objects.

4.3.416 Release_authentication to Date

Each Release_authentication has release_date defined by exactly one Date object. Each Date defines release_date for zero, one, or more Release_authentication objects.

4.3.417 Release_authentication to Identifier

Each Release_authentication has authentication defined by exactly one Identifier object. Each Identifier defines authentication for zero, one, or more Release_authentication objects.

4.3.418 Release_authentication to Person

Each Release_authentication has person_responsible defined by zero or one Person objects. Each Person defines person_responsible for zero, one, or more Release_authentication objects.

4.3.419 Retrofit_usage to Notation

Each Retrofit_usage has retrofit_description defined by zero or one Notation objects. Each Notation defines retrofit_description for zero, one, or more Retrofit_usage objects.

4.3.420 Retrofit_usage to Other_reference_document

Each Retrofit_usage has defining_document defined by zero or one Other_reference_document object. Each Other_reference_document defines defining_document for zero, one, or more Retrofit_usage objects.

4.3.421 Retrofit_usage to Text

Each Retrofit_usage has disposition_for defined by exactly one Text object. Each Text defines disposition_-for for zero, one, or more Retrofit_usage objects.

NOTE Text values limited to “add,” Delete,” and “modify” by retrofit_state.

4.3.422 Revision to Date

Each Revision has revision_date defined by zero or one Date objects. Each Date defines revision_date for zero, one, or more Revision objects.

4.3.423 Revision to Identifier

Each Revision has revision_level defined by exactly one Identifier object. Each Identifier defines revision_-level for zero, one, or more Revision objects.

4.3.424 Revision to Other_reference_document

Each Revision has authorizing_documents defined by zero, one, or more Other_reference_document objects. Each Other_reference_document defines authorizing_documents for zero, one, or more Revision objects.

4.3.425 Revision to Person_and_Organization

Each Revision has revision_approval defined by zero or one Person_and_organization objects. Each Person_and_organization defines revision_approval for zero, one, or more Revision objects.

4.3.426 Revision to Text

Each Revision has revision_description defined by zero or one Text objects. Each Text defines revision_-description for zero, one, or more Revision objects.

4.3.427 Security_classification to Date

Each Security_classification has classification_date defined by zero or one Date objects. Each Date defines classification_date for zero, one, or more Security_classification objects.

Each Security_classification has declassification_date defined by zero or one Date objects. Each Date defines declassification_date for zero, one, or more Security_classification objects.

4.3.428 Security_classification to Label

Each Security_classification has item_classification defined by exactly one Label object. Each Label defines item_classification for zero, one, or more Security_classification objects.

Each Security_classification has title_security_classification defined by zero or one Label objects. Each Label defines title_security_classification for zero, one, or more Security_classification objects.

4.3.429 Security_classification to Person_and_organization

Each Security_classification has classifier defined by zero or one Person_and_organization object. Each Person_and_organization defines classifier for zero, one, or more Security_classification objects.

4.3.430 Sequence_effectivity to Identifier

Each Sequence_effectivity has from_effectivity_id defined by exactly one Identifier object. Each Identifier defines from_effectivity_id for zero, one, or more Sequence_effectivity object

Each Sequence_effectivity has thru_effectivity_id defined by zero or one Identifier objects. Each Identifier defines thru_effectivity_id for zero, one, or more Sequence_effectivity objects.

4.3.431 Sequence_effectivity to Measure

Each Sequence_effectivity has quantity_unit_of_measure defined by zero or one Measure objects. Each Measure defines quantity_unit_of_measure for zero, one, or more Sequence_effectivity objects.

4.3.432 Sequence_effectivity to Quantity

Each Sequence_effectivity has total_component_quantity_in_product_configuration defined by zero or one Quantity objects. Each Quantity has total_component_quantity_in_product_configuration defined by zero, one, or more Sequence_effectivity objects.

NOTE This assertion is established through quantity_type_select.

4.3.433 Shape_model to Geometric_validation_property

Each Shape_model has validation_properties defined by zero, one, or more Geometric_validation_property objects. Each Geometric_validation_property defines validation_properties for zero, one, or more Shape_-model objects.

4.3.434 Shape_model to Geometry

Each Shape_model has part_shape_representation defined by exactly one Geometry object. Each Geometry defines part_shape_representation for zero, one, or more Shape_model objects.

4.3.435 Shaded_shape_model to Shape_model

Each Shaded_shape_model has part_shape_views defined by one or more Shape_model objects. Each Shape_model defines part_shape_views for zero, one, or more Shape_model objects.

4.3.436 Sheet to Configuration

Each Sheet has sheet_configuration defined by zero or one Configuration objects. Each Configuration defines sheet_configuration for zero, one, or more Sheet objects.

NOTE There are rules related to the Sheet object that requires the existence of at least one of the attributes.

4.3.437 Sheet to Element_identification

Each Sheet has sheet_identifier defined by zero or one Element_identification objects. Each Element_-identification defines sheet_identifier for zero, one, or more Sheet objects.

NOTE There are rules related to the Sheet object that requires the existence of at least one of the attributes.

4.3.438 Sheet to Label

Each Sheet has sheet_size defined by zero or one Label objects. Each Label defines sheet_size for zero, one, or more Sheet objects.

NOTE There are rules related to the Sheet object that requires the existence of at least one of the attributes.

4.3.439 Sheet to Product_presentation

Each Sheet has presentation_format defined by zero or one Product_presentation objects. Each Product_-presentation defines presentation_format for zero, one, or more Product_presentation objects.

NOTE There are rules related to the Sheet object that requires the existence of at least one of the attributes.

4.3.440 Simple_list_of_elements to Data_definition_exchange_simple_entry

Each Simple_list_of_elements has element_entries defined by one or more Data_definition_exchange_-simple_entry objects. Each Data_definition_exchange_simple_entry defines element_entries for zero, one, or more Simple_list_of_elements.

4.3.441 Simple_list_of_files to Accessed_file

Each Simple_list_of_files has file_entries defined by zero or one or more Accessed_file objects. Each Accessed_file defines file_entries for zero, one, or more Simple_list_of_files objects.

4.3.442 Simple_list_of_files to Exchange_file

Each Simple_list_of_files has file_entries defined by zero, one, or more Exchange_file objects. Each Exchange_file defines file_entries for zero, one, or more simple_list_of_files objects.

NOTE At least one Accessed_file or Exchange_file must be in the file_entries for a simple_list_of_files.

4.3.443 Size_characteristics to An_integer

Each Size_characteristics has byte_size defined by zero or one An_integer objects. Each An_integer defines byte_size for zero, one, or more Size_characteristics objects.

NOTE There are rules related to the Size_characteristics object that requires the existence of at least one of the attributes.

4.3.444 Size_characteristics to Label

Each Size_characteristics has sheet_size defined by zero or one Label objects. Each Label defines sheet_size for zero, one, or more Size_characteristics objects.

NOTE There are rules related to the Size_characteristics object that requires the existence of at least one of the attributes.

4.3.445 Size_characteristics_internal_divisions to An_integer

Each Size_characteristics_internal_divisions has internal_division_count defined by zero or one An_integer objects. Each An_integer defines internal_division_count for zero, one, or more Size_characteristics_-internal_divisions objects.

NOTE There are rules related to the Size_characteristics_internal_divisions object that requires the existence of at least one of the attributes.

4.3.446 Size_characteristics_internal_divisions to Label

Each Size_characteristics_internal_divisions has internal_division_type defined by zero or one Label objects. Each Label defines internal_division_type for zero, one, or more Size_characteristics_internal_-divisions objects.

NOTE There are rules related to the Size_characteristics_internal_divisions object that requires the existence of at least one of the attributes.

4.3.447 Size_characteristics_sheet_across_file to An_integer

Each Size_characteristics_sheet_across_file has number_of_frames_for_sheet defined by zero or one An_-integer objects. Each An_integer defines number_of_frames_for_sheet for zero, one, or more Size_-characteristics_sheet_across_file objects.

NOTE 1 There are rules related to the Size_characteristics_sheet_across_file object that requires the existence of at least one of the attributes.

Each Size_characteristics_sheet_across_file has frame_number_comprising_this_file defined by zero or one An_integer objects. Each An_integer defines frame_number_comprising_this_file for zero, one, or more Size_characteristics_sheet_across_file objects.

NOTE 2 There are rules related to the Size_characteristics_sheet_across_file object that requires the existence of at least one of the attributes.

4.3.448 Source_information_type to Identifier

Each Source_information_type has source_code defined by zero or one Identifier objects. Each Identifier defines source_code for zero, one, or more Source_information_type objects.

NOTE There are rules related to the Source_information_type object that requires the existence of at least one of the attributes.

4.3.449 Source_information_type to Text

Each Source_information_type has type_of_coding_scheme defined by zero or one Text objects. Each Text defines type_of_coding_scheme for zero, one, or more Source_information_type objects.

Each Source_information_type has source_description defined by zero or one Text objects. Each Text defines source_description for zero, one, or more Source_information_type objects.

NOTE There are rules related to the Source_information_type object that requires the existence of at least one of the attributes.

4.3.450 Source_location to Text

Each Source_location has path_information defined by one Text object. Each Text defines path_-infrormation for zero, one, or more Source_location objects.

Each Source_location has storage_node_identification defined by one Text object. Each Text defines storage_node_identification for zero, one, or more Source_location objects.

4.3.451 Special_condition to Label

Each Special_condition has code defined by zero or one Label objects. Each Label defines code for zero, one, or more Special_condition objects.

NOTE There are rules related to the Special_condition object that requires the existence of at least one of the attributes.

4.3.452 Special_condition to Text

Each Special_condition has type_of_coding_scheme defined by zero or one Text objects. Each Text defines type_of_coding_scheme for zero, one, or more Special_condition objects.

Each Special_condition has description defined by zero or one Text objects. Each Text defines description for zero, one, or more Special_condition objects.

NOTE There are rules related to the Special_condition object that requires the existence of at least one of the attributes.

4.3.453 Specified_part_in_assembly_tree to Assembly_relationship

Each Specified_part_in_assembly_tree has higher_assembly defined by exactly one Assembly_relationship object. Each Assembly_relationship defines higher_assembly for zero, one, or more Specified_part_in_-assembly_tree objects.

4.3.454 Specified_part_in_assembly_tree to Quantified_part_usage_in_-assembly

Each Specified_part_in_assembly_tree has sub_assembly defined by exactly one Quantified_part_usage_-in_assembly object. Each Quantified_part_usage_in_assembly defines sub_assembly for zero, one, or more Specified_part_in_assembly_tree objects.

4.3.455 Status to Identifier

Each Status has status_code defined by exactly one Identifier object. Each Identifier defines status_code for zero, one, or more Status objects.

4.3.456 Status to Text

Each Status has status_code_basis defined by zero or one Text objects. Each Text defines status_code_basis for zero, one, or more Status objects.

4.3.457 Stock_material to Item

Each Stock_material has material defined by zero or one Item objects. Each Item defines material for zero, one, or more Stock_material objects.

NOTE There are rules related to the Stock_material object that requires the existence of at least one of the attributes.

4.3.458 Stock_material to Stock_size

Each Stock_material has size defined by zero or one Stock_size objects. Each Stock_size defines size for zero, one, or more Stock_material objects.

NOTE There are rules related to the Stock_material object that requires the existence of at least one of the attributes.

4.3.459 Stock_size to Stock_size_parameters

Each Stock_size has parameters_for defined by one or many Stock_size_parameters object. Each Stock_-size_parameters defines parameters_for for zero, one, or more Stock_size objects.

4.3.460 Stock_size to Text

Each Stock_size has catalog defined by zero or one Text objects. Each Text defines catalog for zero, one, or more Stock_size objects.

NOTE 1 This assertion is established through stock_size_classification.

Each Stock_size has cross_section defined by zero or one Text objects. Each Text defines cross_section for zero, one, or more Stock_size objects.

NOTE 2 This assertion is established through stock_size_cross_section.

4.3.461 Stock_size_parameter to Label

Each Stock_size_parameter has name_of_parameter defined by zero or one label objects. Each Label defines name_of_parameter for zero, one, or more Stock_size_parameters objects

NOTE There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

4.3.462 Stock_size_parameters to Number_with_units

Each Stock_size_parameters has width defined by zero or one Number_with_units objects. Each Number_-with_units defines width for zero, one, or many Stock_size_parameters objects.

NOTE 1 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has a_length defined by zero or one Number_with_units objects. Each Number_with_units defines a_length for zero, one, or more Stock_size_parameters objects.

NOTE 2 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has radius defined by zero or one Number_with_units objects. Each Number_-with_units defines radius for zero, one, or more Stock_size_parameters objects.

NOTE 3 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has diameter defined by zero or one Number_with_units objects. Each Number_with_units defines diameter for zero, one, or more Stock_size_parameters objects.

NOTE 4 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has wall_thickness defined by zero or one Number_with_units objects. Each Number_with_units defines wall_thickness for zero, one, or more Stock_size_parameters objects.

NOTE 5 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has height defined by zero or one Number_with_units objects. Each Number_-with_units defines height for zero, one or more Stock_size_parameters objects.

NOTE 6 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

Each Stock_size_parameters has numeric_parameter defined by zero or one Number_with_units objects. Each Number_with_units defines numeric_parameter for zero, one, or more Stock_size_parameters objects.

NOTE 7 There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

4.3.463 Stock_size_parameters to Text

Each Stock_size_parameters has textual_parameters defined by zero or one Text objects. Each Text defines textual_parameters for zero, one, or more Stock_size_parameters objects.

NOTE There are rules related to the Stock_size_parameters object that require the existence of at least one of the attributes.

4.3.464 String_location_definition to Load_point

Each String_location_definition has string_load_point defined by exactly one Load_point object. Each Load_point defines string_load_point for zero, one, or more String_location_definition objects.

4.3.465 System_declaration to Identifier

Each System_declaration has system_element_identifier defined by zero or one Identifier objects. Each Identifier defines system_element_identifier for zero, one, or more System_declaration objects.

NOTE 1 There are rules related to the system_declaration object that requires the existence of at least one of the attributes.

Each System_declaration has system_related_element_identifier defined by zero or one Identifier objects. Each Identifier defines system_related_element_identifier for zero, one, or more System_declaration objects.

NOTE 2 There are rules related to the system_declaration object that requires the existence of at least one of the attributes.

Each System_declaration has system defined by zero or one Identifier objects. Each Identifier defines system for zero, one, or more System_declaration objects.

NOTE 3 There are rules related to the system_declaration object that requires the existence of at least one of the attributes.

4.3.466 System_declaration to Text

Each System_declaration has creating_interface defined by zero or one Text objects. Each Text defines creating_interface for zero, one, or more System_declaration objects.

NOTE 1 There are rules related to the system_declaration object that requires the existence of at least one of the attributes.

Each System_declaration has operating_system defined by zero or one Text objects. Each Text defines operationg_system for zero, one, or more System_declaration objects.

NOTE 2 There are rules related to the system_declaration object that requires the existence of at least one of the attributes.

4.3.467 System_destination to Company

Each System_destination has target_company defined by exactly one Company object. Each Company defines target_company for zero, one, or more System_destination objects.

4.3.468 System_destination to Company_code

Each System_destination has target_company_code defined by zero or one Company_code objects. Each Company_code defines target_company_code for zero, one, or more System_destination objects.

4.3.469 System_destination to Person

Each System_destination has target_person defined by zero or one Person objects. Each Person defines target_person for zero, one, or more System_destination objects.

4.3.470 Tabulation to Column_header

Each Tabulation has tabulation_columnar_headings defined by zero, one, or more Column_header objects. Each Column_header defines tabulation_columnar_headings for zero, one, or more Tabulation objects.

4.3.471 Tabulation to Columnar_data_content_holder

Each Tabulation has tabulation_columns defined by one or more Columnar_data_content_holder objects. Each Columnar_data_content_holder defines tabulation_columns for zero, one, or more Tabulation objects.

4.3.472 Tabulation to Group_parent_column_header

Each Tabulation has tabulation_group_parent_headings defined by zero, one, or more Group_parent_header objects. Each Group_parent_header defines tabulation_group_parent_headings for zero, one, or more Tabulation objects.

4.3.473 Tabulation to Group_parent_content_holder

Each Tabulation has parent_columns defined by zero, one, or more Group_parent_column_holder objects. Each Group_parent_column_holder defines parent_columns for zero, one, or more Tabulation objects.

4.3.474 Tabulation to Group_parent_header

Each Tabulation has group_parent_headings defined by zero, one, or more Group_parent_header objects. Each Group_parent_header defines group_parent_headings for zero, one, or more Tabulation objects.

4.3.475 Tabulation to Tabulation_header

Each Tabulation has tabulation_headings defined by zero, one, or more Tabulation_header objects. Each Tabulation_header defines tabulation_headings for zero, one, or more Tabulation objects.

4.3.476 Tabulation_entry to Text

Each Tabulation_entry has column_information defined by one or more Text objects. Each Text defines column_information for zero, one, or more Tabulation_entry objects.

4.3.477 Tabulation_header to Field_title

Each Tabulation_header has single_tabulation_header_titles defined by one or more Field_title objects. Each Field_title defines single_tabulation_header_titles for zero, one, or more Tabulation_header objects.

4.3.478 Tdp_element_and_item_association to Item

Each Tdp_element_and_item_association has related_item defined by exactly one Item object. Each Tdp_-element_and_item_association defines related_item for zero, one, or more Item objects.

4.3.479 Tdp_element_and_item_association to Tdp_element

Each Tdp_element_and_item_association has related_tdp_element defined by exactly one Tdp_Element object. Each Tdp_element defines related_tdp_element for zero, one, or more Tdp_element_and_item_-association objects.

4.3.480 Tdp_element_list_item to Element_identification

Each Tdp_element_list_item has tdp_element_item defined by exactly one Element_identification object. Each Element_identification defines tdp_element_item for zero, one, or more Tdp_element_list_item objects.

4.3.481 Tdp_element_list_item to Identifier

Each Tdp_element_list_item has reference_code defined by zero or one Identifier objects. Each Identifier defines reference_code for zero, one, or more Tdp_element_list_item objects.

4.3.482 Tdp_element_parent_to_item_child_relationship to Element_-identification

Each Tdp_element_parent_to_item_child_relationship has parent defined by exactly one Element_-identification object. Each Element_identification defines parent for zero, one, or more Tdp_element_-parent_to_item_child_relationship objects.

4.3.483 Tdp_element_parent_to_item_child_relationship to Item_identification

Each Tdp_element_parent_to_item_child_relationship has child defined by exactly one Item_identification object. Each Item_identification defines child for zero, one, or more Tdp_element_parent_to_item_child_-relationship objects.

4.3.484 Tdp_element_parent_to_tdp_element_child_relationship to Element_-identification

Each Tdp_element_parent_to_tdp_element_child_relationship has parent defined by exactly one Element_-identification object. Each Element_identification defines parent for zero, one, or more Tdp_element_-parent_to_tdp_element_child_relationship objects.

4.3.485 Tdp_element_parent_to_tdp_element_child_relationship to Item_-identification

Each Tdp_element_parent_to_tdp_element_child_relationship has child defined by exactly one Element_-identification object. Each Element_identification defines child for zero, one , or more Tdp_element_parent_-to_tdp_element_child_relationship objects.

4.3.486 Tdp_indentured_item to Item_identification

Each Tdp_indentured_item has top_indenture defined by exactly one Item_identification object. Each Item_-identification defines top_indenture for zero, one, or more Tdp_indentured_item objects.

4.3.487 Tdp_indentured_tdp_element to Element_identification

Each Tdp_indentured_tdp_element has top_indenture defined by exactly one Element_identification object. Each Element_identification defines top_indenture for zero, one, or more Tdp_indentured_tdp_element objects.

4.3.488 Time to A_real

Each Time has second defined by zero or one A_real objects. Each A_real defines second for zero, one, or more Time objects.

4.3.489 Time to An_integer

Each Time has hour defined by exactly one An_integer object. Each An_integer defines hour for zero, one, or more Time objects.

Each Time has minute defined by zero or one An_integer objects. Each An_integer defines minute for zero, one, or more Time objects.

4.3.490 Time to Text

Each Time has zone defined by exactly one Text object. Each Text defines zone for zero, one, or more Time objects.

4.3.491 Time_interval_effectivity to Date

Each Time_interval_effectivity has primary_bound defined by exactly one Date object. Each Date defines primary_bound for zero, one, or more Time_interval_effectivity objects.

NOTE This assertion is established through date_or_event.

Each Time_interval_effectivity has secondary_bound defined zero or one Date objects. Each Date defines secondary_bound for zero, one, or more Time_interval_effectivity objects.

4.3.492 Time_interval_effectivity to Event

Each Time_interval_effectivity has primary_bound defined by exactly one Event object. Each Event defines primary_bound for zero, one, or more Time_interval_effectivity objects.

NOTE This assertion is established through date_or_event.

Each Time_interval_effectivity has secondary_bound defined zero or one Event objects. Each Event defines secondary_bound for zero, one, or more Time_interval_effectivity objects.

4.3.493 Time_interval_effectivity to Label

Each Time_interval_effectivity has interval_name defined by exactly one Label object. Each Label defines interval_name for zero, one, or more Time_interval_effectivity objects.

4.3.494 Time_interval_effectivity to Number_with_units

Each Time_interval_effectivity has duration defined by zero or one Number_with_units objects. Each Number_with_units defines duration for zero, one, or more Time_interval_effectivity objects.

4.3.495 Volume to Number_with_units

Each Volume has a_value defined by exactly one Number_with_units object. Each Number_with_units defines a_value for zero, one, or more Volume objects.

4.3.496 Weight to A_real

Each Weight has a_value defined by exactly one A_real object. Each A_real defines a_value for zero, one, or more Weight objects.

4.3.497 Weight to Measure

Each Weight has unit_of_measure defined by exactly one Measure object . Each Measure defines unit_of_-measure for zero, one, or more Weight objects.

4.3.498 Weight to Text

Each Weight has derivation_method defined by zero or one Text objects. Each Text defines derivation_-method for zero, one, or more Weight objects.

5 Application interpreted model

5.1 Mapping table

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM resource constructs (see annex A).

A number of application objects are common across all UoFs except for Presentation (PRES) and, in some cases, Reference Document (RD). The mappings for these common application objects are contained in Table 4 and are not repeated in the other mapping tables. Table 2 can be used to determine which of the application objects in Table 4 do not belong to the Reference Document UoF.

NOTE 1 References and notes that are found in the application element column of the mapping tables are described at the end of Mapping Table 14.

The mapping table is organized in five columns. The contents of these five columns are:

Column 1) Application element: Name of and application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lowercase.

Column 2) AIM element: Name of an AIM element as it appears in the AIM (see Annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name>.<attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion maps to the entire reference path.

Column 3) Source: For those AIM elements that are interpreted from the integrated resources, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part. For those elements that are directly incorporated from an application interpreted construct (AIC), this is the AIC reference.

Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. The expanded names of the rules are listed after the table.

Column 5) Reference path: To describe fully the mapping of an application object, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths and the relationships between AIM elements the following notational conventions apply:

a) [] : multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

b) () : multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

c) {} : enclosed section constrains the reference path to satisfy an information requirement;

d) <> : enclosed section constrains at one or more required reference path;

e) || : enclosed section constrains the supertype entity;

f) -> : attribute references the entity or select type given in the following row;

g) <- : entity or select type is referenced by the attribute in the following row;

h) [i] : attribute is an aggregation of which a single member is given in the following row;

i) [n] : attribute is an aggregation of which member n is given in the following row;

j) => : entity is a supertype of the entity given in the following row;

k) <= : entity is a subtype of the entity given in the following row;

l) = : the string, select, or enumeration type is constrained to a choice or value;

m) \ : the reference path expression continues on the next line.

Table 4 — Mapping table for common

Application element	AIM element	Source	Rules	Reference path
A_NUMBER	number	11		
A_REAL	real	11		
ALTERNATE_ELEMENT_- IDENTIFICATION	product_definition_formation	41		{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name='document'} <{product_definition_formation product_or_formation_or_definition=product_definition_formation product_or_formation_or_definition<- document_product_association.related_product document_product_association document_product_association.name='equivalence'} {product_definition_formation product_or_formation_or_definition=product_definition_formation product_or_formation_or_definition<- document_product_association.related_product document_product_association document_product_association.relating_document-> document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'}>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_element_identification to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_element_identification to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_element_identification to element_type (element_classifiations)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category
alternate_element_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_element_identification to other_reference_document (outstanding_changes)	PATH			(product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='outstanding changes'} ship.related_product_definition_formation-> product_definition_formation) (product_definition_formation document_reference_item=product_definition_formation document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='outstanding changes')} document_reference.assigned_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_element_identification to text (title)	PATH			product_definition_formation product_definition_formation._of_product-> product product.name
AN_INTEGER	integer	11		
APPROVAL	applied_approval_assignment	232		applied_approval_assignment<= approval_assignment
approval to date (approval_date) #1: Used when multiple people are assigned to one approval instance.	PATH		2	applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select date_time_select=(date_and_time)(date) (date_and_time)(date) #1: <applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval<- approval_person_organization.authorized_approval approval_person_organization date_and_time_item=approval_person_organization date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date_and_time-> date_and_time>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
approval to person_organization (person_organization) #1: for single person signoff #2: for multiple person signoff - an unique approval_relationship is used for each person signing off	PATH		3	#1: (applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval->approval<- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization->person_organization_select person_organization_select=person_and_organization person_and_organization) #2: (applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval->approval<- approval_relationship.relating_approval approval_relationship {approval_relationship.name='decomposition'} approval_relationship.related_approval->approval<- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization->person_organization_select person_organization_select=person_and_organization person_and_organization)
approval to text (purpose_of)	PATH			applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval->approval approval.level

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
approval to text (status) #1: Provides alias name for approval status.	PATH			applied_approval_assignment<= approval_assignment approval_assignment.status-> approval_status #1: <identification_item=approval_status identification_item<- applied_identificaiton_assignment.item[i] applied_identification_assignment=> identification_assignment {identification_assignment..role-> identification_role identification_role.name='alias'}>
CERTIFICATION	certification	41		
certification to approval (approvals) #1: approval of the certification. #2: approval to apply the certification to the product data.	PATH			#1: (certification approval_item =certification) #2: (certification<- applied_certification_assignment.assigned_certification applied_certification_assignment approval_item=applied_certification_assignment) approval_item<- applied_approval_assignement.items[i] applied_approval_assignment <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
certification to date (creation_date)	PATH			#1: (certification date_item=certification date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role-> date_role (date_role.name) (date_role.name='creation')} date_assignment.assigned_date-> date) #2: (certification date_and_time_item=certification date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role-> date_time_role (date_time_role.name) (date_time_role.name='creation')} date_and_time_assignment.assigned_date-> date_and_time)
certification to label (name)	PATH			certification certification.name
certification to text (description)	PATH			certification certification.purpose
CHANGE_IDENTIFICATION	executed_action	41	4	{executed_action<= action action.description='change identification'}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to approval (revision_authorization_-identifications) #1: approval of the change activity. #2: approval to apply the change to the product data.	PATH			#1: (executed_action approval_item =executed_action) #2: (executed_action<= action<- applied_action_assignment.assigned_action applied_action_assignment approval_item=applied_action_assignment) approval_item<- applied_approval_assignement.items[i] applied_approval_assignment <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role>
change_identification to change_-type (change_code)	PATH			executed_action<= action action.name {(action.name) (action.name='original') (action.name='original with change') (action.name='revision') (action.name='revision with change') (action.name='change') (action.name='other')}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to contract (revision_authorization_-identifications)	PATH			executed_action<= action contract_item=action contract_item<- applied_contract_assignment.items[i] applied_contract_assignment<= contract_assignment {contract_assignment<- role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='contracted element') (object_role.name= 'preparing contract')} contract_assignment.assigned_contract-> contract

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to date (change_date) #1: just date #2: date and time	PATH			#1: (executed_action date_item=executed_action date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date-> date {date_assignment.role-> date_role date_role.name='change date'}) #2: (executed_action date_and_time_item=executed_action date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date-> date_and_time {date_and_time_assignment.role-> date_time_role date_time_role.name='change date'})

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to date (issue_date) #1: just date #2: date and time	PATH			#1: (executed_action date_item=executed_action date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date-> date {date_assignment.role-> date_role (date_role.name) (date_role.name='issue date') (date_role.name=sequence date')}) #2: (executed_action date_and_time_item=executed_action date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date-> date_and_time {date_and_time_assignment.role-> date_time_role (date_time_role.name) (date_time_role.name='issue date') (date_time_role.name='sequence date')})

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to date (revision_date) #1: just date #2: date and time	PATH			#1: (executed_action date_item=executed_action date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date->date {date_assignment.role->date_role date_role.name='revision date'}) #2: (executed_action date_and_time_item=executed_action date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date->date_and_time {date_and_time_assignment.role->date_time_role date_time_role.name='revision date'})
change_identification to identifier (change_level)	PATH			executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='change identification level'} action_relationship.related_action->action {action.description='change'} action.name

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to identifier (issue_level) #1: Use to capture issue or sequence string value. #2: Use to capture explicit relationship between two sequences or issue. Could use this to derive a sequence or issue level based on order of the product_definition_-formation_relationships.	PATH			#1:<executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='change identification level'} action_relationship.related_action-> action {(action.description='issue') (action.description='sequence')} action.name> #2:<executed_action action<- applied_action_assignment.assigned_action action_assignment=> {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} applied_action_assignment applied_action_assignment.items[i]-> action_item action_item=product_definition_formation product_definition_formation product_definition_formation_relationship.related_product_definition_formation {product_definition_formation_relationship.name='sequence'}>
change_identification to identifier (revision_authorization_-identifications)	PATH			executed_action<= action id_attribute_select=action id_attribute_select<- id_attribute.identified_item id_attribute id_attribute.attribute_value

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to identifier (revision_level) #1: For versioning products (for example, parts and documents) #2: For versioning files	PATH			<executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='change identification level'} action_relationship.related_action-> action {action.description=’revision'} action.name> executed_action<= action<- action_assignment.assigned_action action_assignment=> {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} applied_action_assignment applied_action_assignment.items[i]-> action_item #1: (action_item=product_definition_formation product_definition_formation product_definition_formation.id) #2: (action_item=document_file document_file identification_item=document_file identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment {identification_assignment.role-> identification_role.name=’version’} identification_assignment.assigned_id)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to other_- reference_document (revision_authorization_- identifications) #1:reference full document #2: reference portion of document	PATH			executed_action document_reference_item=executed_action document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorization identification') (object_role.name='change rationale')} document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to persion_- and_organization (assigned_to) #1: for persion and organization #2: for just organization #3: assigned activity #4: assigned activity relative to applying the change	PATH			#1: (executed_action<= #3:(action person_and_organization_item=action) #4:(action<- applied_action_assignment.assigned_action aapplied_action_assignment persion_and_organization_item=applied_action_assignment) person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=persion_and_organization person_organization_select) #2: (executed_action #3:(action organization_item=action) #4:(action<- applied_action_assignment.assigned_action aapplied_action_assignment organization_item=applied_action_assignment) organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
change_identification to status (action_status)	PATH			executed_action<- action_status.assigned_action action_status

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to text (change_description)	PATH			executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='change identification level'} action_relationship.related_action-> action action.chosen_method-> action_method action_method.description

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
change_identification to text (revision_description)	PATH			<executed_action<= action<- applied_action_assignment.assigned_action action_assignment=> {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} applied_action_assignment applied_action_assignment.items[i]-> action_item action_item=product_definition_formation product_definition_formation product_definition_formation.description> <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name=’change identification level’} action_relationship.related_action-> action action.chosen_method-> action_method action_method.description>
COMPANY	organization	41		
company to label (name)	PATH			organization organization.name

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
company to text (address)	PATH			organization<- organizational_address.organizations[i] organizational_address<= address
COMPANY_CODE	organization	41		
company_code to identifier (code) #1: Only one code for organization and no type of administrator information or code type needed. #2: One or more code for organization needed or type of code or code administrator information needed.	PATH			#1 : organization.id #2 : organization<- identification_item=organization identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment identificaiton_assignment.assigned_id {identification_assignment.role->identification_role (identification_role.name='alias id') (identification_role.name=’alias’)}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
company_code to company (code_administrator)	PATH			organization<- identification_item=organization identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment {identification_assignment.role-> identification_role (identification_role.name='alias id') (identification_role.name=’alias’)} {(identification_role.description) (identification_role.description=’CAGE’) (identification_role.description=‘commercial and government entity’) (identification_role.description=‘CEC’) (identification_role.description=‘contractor establishment code’) (identification_role.description=‘DUNS’) (identification_role.description=‘data universal numbering system’) (identification_role.description=‘EIN’) (identification_role.description=‘employer identification number’) (identification_role.description=‘PASS’) (identification_role.description=‘procurement automated source system’)} organization_item=identification_assignment organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name=’alias id administrator’) (organization_role.name=’alias scope’)} organization_assignment.assigned_organization-> organization

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
company_code to text (type_of_code)	PATH			organization<- identification_item=organization identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment <identification_assignment.role-> identification_role {(identification_role.description) (identification_role.description='CAGE') (identification_role.description='CEC') (identification_role.description='DUNS') (identification_role.description='EIN') (identification_role.description='PASS')} {(identification_role.name='alias id) (identification_role.name-'alias')}> <organization_item=identification_assignment organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.role-> organization_role [description_attribute_select=organization_role description_attribute_select<- description_attribute.described_item description_attribute {(description_attribute.attribute_value) (description_attribute.attribute_value='Commercial and Government Entity') (description_attribute.attribute_value='Contractor Establishment Code') (description_attribute.attribute_value='Data Universal Numbering System') (description_attribute.attribute_value='Employer Identification Number') (description_attribute.attribute_value='Procurement Automated Source System')}] [organization_role.name='alias scope']>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
CONFIGURATION	property_definition	41		
configuration to contract (preparing_contracts) #1: use for applying contract to property of a part or document. #2: use for applying contract to part definition or document represenation #3: use for applying contract to version of part or document version.	PATH			property_definition #1: (contract_item=property_definition) (property_definition.definition->) characterized_definition characterised_definition==characterized_product_definition characterized_product_definition= product_definition product_defintion #2: (contract_item=product_definition) #3: (product_definition.formation->) product_definition_formation contract_item=product_definition_formation)) contract_item<- applied_contract_assignment.items[i] applied_contract_assignment<= contract_assignment {contract_assignment<- role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role->) object_role (object_role.name) (object_role.name=’contracted element’) (object_role.name= ‘preparing contract’)} contract_assignment.assigned_contract->) contract

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
configuration to distribution_notice (distribution_authorizations)	PATH			property_definition property_definition.definition-> characterized_definition characterised_definition =characterized_product_definition characterized_product_definition=product_definition product_defintion (approval_item=product_definition) (product_definition.formation-> product_definition_formation approval_item=product_definition_formation) approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='distribution notice'} approval_assignment.assigned_approval-> approval

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
configuration to label (data_usage_rights)	PATH		17	property_definition approval_item=property_definition approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='data usage rights'} approval_assignment.assigned_approval-> approval approval.level <approval.status-> approval_status approval_status.name>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
configuration to product_-configuration (end_item_system_designation) #1: when configuration is for an item/part #2: when configuration is for a group of product data / document where the usage context within the configuration item is required. #3: when configuration is for a group of product data / document, where no usage context within the configuration is required. #4: When usage is based on item being used or subordinate to. #5: When usage is based on item being resultant of other items or being the parent of the usage context..	PATH			#1, #2: (property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition<- #4: (product_definition_relationship.related_product_definition) #5: (product_definition_relationship.relating_product_definition) product_definition_relationship<- product_definition_effectivity.usage product_definition_effectivity=> configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item) #3: property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition product_definition.formation-> product_definition_formation product_definition_formation=configuration_design_item configuration_design_item<- configuration_design.design configuration_design {name_attribute_select=configuration_design name_attributre_select<- name_attribute.named_item name_attribute name_attribute.attribute_value='documentation configuration'} configuration_design.configuration-> configuration_item)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
configuration to release_-authentication (release_authorizations) #1 use for applying release_-authentication to property of a part or document. #2 use for applying release_-authentication to part definition or document representation. #3 use for applying release_-authentication to version of part or document version.	PATH		17	property_definition approval_item=property_definition approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='release authentication'} approval_assignment.assigned_approval-> approval
configuration to security_-classifications (security_identifications)	PATH			property_definition (security_classification_item=property_definition) (property_definition.definition-> characterized_definition characterised_definition =characterized_product_definition characterized_product_definition=product_definition product_defintion (security_classification_item=product_definition) (product_definition.formation-> product_definition_formation security_classification_item=product_definition_formation)) security_classification_item<- applied_security_classification_assignment.items[i] applied_security_classification_assignment<= security_classification_assignment security_classification_assignment.assigned_security_classification-> security_classification

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
CONTRACT	contract	41		
contract to approval (approvals)	PATH			contract approval_item=contract approval_item<- applied_approval_assignement.items[i] applied_approval_assignment <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
contract to date (creation_date) #1: If the date includes date and time #2: If the date only includes a date	PATH			contract #1: (date_and_time_item=contract date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role-> date_time_role (date_time_role.name) (date_time_role.name='creation')} date_and_time_assignment.assigned_date_and_time-> date_and_time) #2: (date_item=contract date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment date_assignment.role-> date_role (date_role.name) (date_role.name='creation')} date_assignment.assigned_date-> date)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
contract to identifier (contract_data_requirements_list)	PATH			contract<- contract_assignment.assigned_contract contract_assignment=> {contract_assignment<- role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='contract document'} applied_contract_assignment applied_contract_assignment.items[i]-> contract_item contract_item=product_definition_formation product_definition_formation {product_definition_formation-> product<- product_related_product_category.products product_related_product_category<= product_category.name='document'} product_or_formation_or_definition=product_definition_formation product_or_formation_or_definition<- document_product_association.related_product document_product_association {document_product_association=> document_product_equivalence} document_product_association.relating_document-> document<- document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element='contract data requirements list'} document_usage_constraint.subject_element_value <document_usage_constraint<- document_usage_constraint_assignment.assigned_document_usage document_usage_constraint_assignment {document_usage_constriant_assignment=> applied_document_usage_constraint_assignment} document_usage_constraint_assignment.item[i]-> document_reference_item

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				(document_reference_item=product_definition) (document_reference_item=product_definition_formation) (document_reference_item=applied_document_reference)>
contract to identifier (contract_number)	PATH			contract-> contract.name
contract to identifier (data_item_description_- identification)	PATH			contract<- contract_assignment.assigned_contract contract_assignment=> {contract_assignment<- role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='reference from contract'} applied_contract_assignment applied_contract_assignment.items[i]-> contract_item contract_item=product_definition_formation product_definition_formation {product_definition_formation-> product<- product_related_product_category.products product_related_product_category<= product_category.name='document'} {product_definition_formation-> product<- product_related_product_category.products product_related_product_category<= product_category.name='data item description list'}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
contract to person_and_- organization (affected_organization) #1: person_and_organization #2: just organization	PATH			#1: (contract person_and_organization_item=contract person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=persion_and_organization person_organization_select) #2: (contract organization_item=contract organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
DATE	(date)(date_and_time)	41		
date to an_integer (day)	PATH			date=> calendar_date calendar_date.day_component
date to an_integer (month)	PATH			date=> calendar_date calendar_date.month_component
date to an_integer (week)	PATH			date=> week_of_year_and_day_date week_of_year_and_day_date.week component

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
date to an_integer (year)	date.year_component	41		
date to time (specific_time)	PATH			date<- date_and_time.date_component date_and_time date_and_time.time_component local_time
DESIGN_AUTHORITY	(person_and_organization_assignment) (organization_assignment)	41		
design_authority to company (design_activity_identification) #1: just organization #2: organization with person	#1,#2: PATH			#1:(organization_assignment organization_assignment.assigned_organization->organization) #2:(person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization->person_and_organization person_and_organization.the_organization->organization)
design_authority to company_code (design_activity_code) #1: just organization #2: organization with person	#1,#2: PATH			#1:(organization_assignment organization_assignment.assigned_organization->organization) #2:(person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization->person_and_organization person_and_organization.the_organization->organization)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
design_authority to label (role) #1: just organization #2: organization with person				#1: (organization_assignment organization_assignment.role->organization_role {(organization_role.name) (organization_role.name='id owner') (organization_role.name='creator')}) #2: (person_and_organization_assignment person_and_organization_assignment.role->person_and_organization_role {(organizaiton_role.name) (person_and_organization_role.name='id owner') (person_and_organization_role.name='creator')})
design_authority to person (person_responsible) #1: just organization (see note 2) #2: organization with person	#2: PATH			#2:(person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization->person_and_organization person_and_organization.the_person->person)
DISTRIBUTION_NOTICE	approval	41		
distribution_notice to label (distribution_code)	PATH		11	approval classification_item=approval classification_item<-applied_classification_assignment.items[i] applied_classification_assignment<=classification_assignment {classification_assignment classifcation_assignment.role->classification_role classification_role.name='distribution notice code'} classification_assignment.assigned_class->group group.name {group=>class}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
distribution_notice to person_and_- organization (distribution_authority)	PATH		11	approval<- approval_person_organization.authorized_approval approval_person_organization {approval_person_organization.role-> approval_role approval_role.name='distribution authorizaton'} approval_person_organization.person_organization-> person_organization_select (person_organization_select=organization) (person_organization_select= person_and_organization)
distribution_notice to text (distribution_statement)	PATH		11	approval approval.level
DOCUMENT_USAGE_- PARAMETER	document_usage_constraint	41		
document_usage_parameter to label (subject)	PATH			document_usage_constraint document_usage_constraint.subject_element_value {(document_usage_constraint.subject _element_value) (document_usage_constraint.subject_element_value='type') (document_usage_constraint.subject_element_value='grade') (document_usage_constraint.subject_element_value='form') (document_usage_constraint.subject_element_value='composition')}
document_usage_parameter to text (value_of)	PATH			document_usage_constraint document_usage_constraint.subject_element_value

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
ELEMENT_IDENTIFICATION #1: This constraint is always applied for an Element_-identification. #2: This constraint is only used when a document_reference or a document_usage_constraint is to be associated with an Element_-identification (product_definition_-formation).	product_definition_formation	41	24	#1: {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name='document'} #2: <{product_definition_formation product_or_formation_or_definition=product_definition_formation product_or_formation_or_definition<- document_product_association.related_product document_product_association document_product_association.name='equivalence'} {product_definition_formation product_or_formation_or_definition=product_definition_formation product_or_formation_or_definition<- document_product_association.related_product document_product_association document_product_association.relating_document-> document document.kind-> document_type document_type.product_data_type='configuration controlled document version'}>
element_identification to alternate_element_identification (alternate_identifications) #1: When alternate identification is for a combination of version and id #2: When alternate identificationn is for supplier id and version of document.	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship #1: ({product_definition_formation_relationship.name='alternate element identification'}) #2:({product_definition_formation_relationship.name='supplied document'}) product_definition_formation_relationship.relatrd_product_definition_formation-> product_definition_formation

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
element_identification to certification (element_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification-> certification
element_identification to change_-identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
element_identification to design_-authority (design_activities) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator')})>
element_identification to element_-type (element_classifiations)	PATH			product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products product_related_product_category<= product_category

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
element_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identification_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
element_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
element_identification to other_- reference_document (outstanding_changes) #1: reference full document #2: reference portion of document	PATH			(product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='outstanding changes'} product_definition_formation_relationship.related_product_definition_formation->) ((product_definition_formation document_reference_item=product_definition_formation) (product_definition_formation.of_product-> product document_reference_item=product) document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='outstanding changes')} document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation)
element_identification to text (title)	PATH			product_definition_formation product_definition_formation._of_product-> product product.name
ELEMENT_TYPE	product_category	41		
element_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='code administrator'} organization_assignment.assigned_organization-> organization
element_type to label (element_code)	PATH			product_category product_category.name
element_type to text (type_of_coding_scheme)	PATH			product_category product_category.description
HEADER	property_definition	41		

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
header to an_integer (sheet_count)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_defintion characterized_product_definition=product-definition product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.defintion property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document parameters'} representation.items[i]-> representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value measure_value=count_measure count_measure {representation_item.name='page count'} {measure_with_unit.unit_component-> unit unit=named_unit named_unit=> context_dependent_unit context_dependent_unit.name='count'}
header to header_configuration_-with_element_identification (header_configuration)	IDENTICAL MAPPING			

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
header to label (languages)	PATH			property_definition property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document languages'} representation.items[i]-> representation_item=> descriptive_representation_item descriptive_representation_item.description
header to size_characteristics (size_of_sheet)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_defintion characterized_product_definition=product-definition product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.defintion property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
header to text (document_abstract)	PATH			property_definition property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document abstract'} representation.items[i]-> representation_item=> descriptive_representation_item descriptive_representation_item.description
header to text (document_keywords)	PATH			property_definition property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document keywords'} representation.items[i]-> representation_item=> descriptive_representation_item descriptive_representation_item.description
HEADER_CONFIGURATION_-WITH_ELEMENT_-IDENTIFICATION	property_definition	41		
header_configuration_with_-element_identification to configuration (data_configuration)	IDENTICAL MAPPING			

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
header_configuration_with_- element_identification to element_- identification (identification)	PATH	41		property_defintion property_definition.definition->characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition product_definition.formation->product_definition_formation
IDENTIFIER	identifier	41		
ITEM	product_definition	41		

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to alternate_item (alternates) #1: If alternate is a substitute with no concern for which is used in an assembly context #2: If alternate is an alternate item with concern for which is used in an assembly context. #3: If alternate is an alternate item with a one for one substitution and allows for two way substitution in a particular assembly context	PATH			#1: (product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- alternate_product_relationship.base alternate_product_relationship alternate_product_relationship.alternate-> product) #2: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.base (assembly_component_usage_substitute) (assembly_component_usage_substitute=> assembly_component_usage_substitute_with_ranking) assembly_component_usage_substitute.substitute-> assembly_component_usage<= product_definition_usage<= product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product) #3: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship<- product_definition_substitute.context_relationship product_definition_substitute product_definition_substitute.substitute_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation_of_product-> product)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to configuration (item_configuration)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='item configuration'}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to file (shape) #1: Item has configuration control of shape file. #2: Item may have configuration control of shape file. #3: Path required when locatoin or placement information needs to be associated with Item's shape.	PATH			#1: (product_definition=> product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i]-> document=> document_file) #2: (product_defintion document_reference_item=product_defintion document_reference_item<- applied_document_reference.item[i] applied_document_reference<= document_reference=> document_file) #3: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> represented_definition represented_defintion=property_defintion property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_object characterized_object=> document_file)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to file (shape_aspects) #1: shape representation of shape aspect through property_definition #2: shape representation of shape aspect directly to representation	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.defintion property_definition=> product_definition_shape<- shape_aspect.of_shape shape_aspect #1: (shape_definition=shape_aspect characterized_definition=shape_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition) #2: (represented_definition=shape_aspect) represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> represented_definition represented_defintion=property_defintion property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_object characterized_object=> document_file

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to item_identification (identification)	PATH			product_definition product_definition.formation-> product_definition_formation

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to label (item_context) #1: Provides alias name for application_context.	PATH			(product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product product.frame_of_references[i]-> product_context<= application_context_element application_context_element.name) (product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product product.frame_of_references[i]-> product_context<= application_context_element application_context_element.frome_of_reference-> application_context application_context.application) ((product_definition product_definition.frame_of_reference->) (product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference->) product_definition_context product_definition_context<= application_context_element application_context_element.name) ((product_definition product_definition.frame_of_reference->) (product_definition<- product_definition_context_asscoaition.definition product_definition_context_association product_definition_context_association.frame_of_reference->) product_definition_context product_definition_context<= application_context_element application_context_element.frame_of_reference->

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context #1: <identification_item=application_context identification_item<- applied_identificaiton_assignment.item[i] applied_identification_assignment=> identification_assignment {identification_assignment..role-> identification_role identification_role.name='alias'}> application_context.application)
item to notation (notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='descriptive properties'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to reference_document_usage (conditions_defined_through_-constrained_document) #1: Multiple constraints parameters that are applied together. #2: Single constraint applied independently	PATH			#1: (product_definition document_reference_item=product_definition document_reference_item<- applied_document_reference.items[i] applied_document_reference <applied_document_reference<= document_reference<- role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='finish condition') (object_role.name='final condition')}> document_reference_item=applied_document_reference document_reference_item<- applied_document_usage_constraint_assignment.items[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment {document_usage_constraint_assignment.role-> document_usage_role (document_usage_role.name) (document_usage_role.name='final conditions') (document_usage_role.name='finish condition')} document_usage_constraint_assignment.assigned_document_usage-> document_usage_constraint) #2: (product_definition document_reference_item=product_definition document_reference_item<- applied_document_usage_constraint_assignment.items[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment {document_usage_constraint_assignment.role-> document_usage_role (document_usage_role.name) (document_usage_role.name='final conditions') (document_usage_role.name='finish condition')}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				document_usage_constraint_assignment.assigned_document_usage->document_usage_constraint)
item to reference_document (conditions_defined_through_-simple_reference)	PATH			product_definition document_item=product_definition document_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='finish condition') (object_role.name='final condition')} document_reference.assigned_document-> document<- {docoument.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to shape_model (shape) #1: Allows ability to apply global uncertainty value to shape model	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition<- {property_definition=> product_definition_shape} property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation=> shape_representation #1: <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to shape_model (shape_aspects) #1: Allows ability to apply global uncertainty value to shape model #2: relates two aspects of shape together #3:shape representation of shape aspect through property_definition #4: shape representation of shape aspect directly to representation #5: Provides abilitiy to assign a document or portion of a document to an aspect of shape or relationship between two aspects of shape. #6: Provide ability to assign an alias to a relationship between two aspects of shape.. #7: shape representation of a relationship between two shape aspects directly to representation	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.defintion property_definition=> product_definition_shape<- shape_aspect.of_shape shape_aspect #2: <shape_aspect<- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship #6:<identification_item=shape_aspect_relationship identification_item<- applied_identificaiton_assignment.item[i] applied_identification_assignment=> identification_assignment {identification_assignment..role-> identification_role identification_role.name='alias'}> #5: <document_reference_item=shape_aspect_relationship document_reference_item<- (applied_document_reference.items[i] applied_document_reference<= document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> shape_aspect_relationship.related_shape_aspect-> shape_aspect> #5: <document_reference_item=shape_aspect document_reference_item<- (applied_document_reference.items[i] applied_document_reference<=

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
				document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage->) document_usage_constraint document_usage_constraint.source->) document> #3: (shape_definition=shape_aspect characterized_definition=shape_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition) #4: (represented_definition=shape_aspect) #7:(shape_aspect<- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship represented_definition=shape_aspect_relationship) represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} #1: <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
item to special_condition (special_conditions)	PATH			product_definition characterized_product_definition=product_definition characterized_product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item representation_item=> descriptive_representation_item
item to weight (item_weight)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition <- {property_definition.name='weight'} property_definition_representation.definition property_ definition_representation property_definition_representation.used_representation-> representation representation.items[i]-> representation_item
LABEL	label	41		
MEASURE	named_unit	41		

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
OTHER_REFERENCE_- DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name='document'} {product_definition_formation<- product_definition product_definition.frame_of_reference-> product_defninition_context<= application_context_element application_context_element.frame_of_reference-> application_context application_context.application='reference document'}
PERSON	person	41		
person to identifier (first_name)	person.first_name	41		
person to identifier (last_name)	person.last_name	41		
person to identifier (middle_name)	person.middle_name	41		
person to identifier (unique_identifier)	person.id	41		
person to text (title)	<person.prefix_titles> <person.suffix_titles>	41		

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
PERSON_AND_- ORGANIZATION	person_organization_select	41		
person_and_organization to company (organization)	PATH			person_organization_select (person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization->) organization) (person_organization_select=organization organization)
person_and_organization to person (person_identification)	PATH			person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_person->) person
person_and_organization to text (electronic_mail_address)	PATH			person_organization_select (person_organization_select=person_and_organization person_and_organization person_and_organization.the_person->) (person_organization_select=person) person<- personal_address.people personal_address<= address address.electronic_mail_address

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
person_and_organization to text (person_address)	PATH			person_organization_select (person_organization_select=person_and_organization person_and_organization person_and_organization.the_person->) (person_organization_select=person) person<- personal_address.people personal_address<= address
PRODUCT_CONFIGURATION	configuration_item	44		
product_configuration to approval (approval_of_configuration)	PATH			configuration_item approval_item=configuration_item approval_item<- applied_approval_assignement.items[i] applied_approval_assignment <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role>
product_configuration to identifier (configuration_id)	PATH			configuration_item configuration_item.id

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
product_configuration to item_-identification (design_item) #1: Configuration based on a particular product version. #2: Configuration based on a particular product view.	PATH			configuration_item<- configuration_design.configuration configuration_design configuration_design.design-> configuration_design_item #1: (configuration_design_item=product_definition_formation product_definition_formation) #2: (configuration_design_item=product_definition product_definition product_definition.formation-> product_definition_formation)
product_configuration to person_-and_organization (assigned_to) #1: person and organization #2: just organization	PATH			#1: (configuration_item persion_and_organization_item=configuration_item person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=persion_and_organization person_organization_select) #2: (configuration_item organization_item=configuration_item organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
product_configuration to product_-model (product_name)	PATH			configuration_item configuration_item.item_concept-> product_concept

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_MODEL	product_concept	44		
product_model to label (model_name)	PATH			product_concept product_concept.id
product_model to project (projects)	PATH			product_concept organizational_project_item=product_concept organizational_project_item<- applied_organizational_project_assignment.items[i] applied_organizational_project_assignment=> organizational_project_assignment organizational_project_assignment.assigned_organizational_project-> organizational_project .
REFERENCE_DOCUMENT	product_definition_formation	41	26 24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_category.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='reference document'}
reference_document to element_-identification (document_identification)	IDENTICAL MAPPING			
REFERENCE_DOCUMENT_-USAGE	document_usage_constraint	41		

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
reference_document_usage to document_usage_paraameter (usage_parameter)	IDENTICAL MAPPING			
reference_document_usage to identifier (usage_code)	PATH			document_usage_constraint document_usage_constraint.subject_element_value {document_usage_constraint.subject_element='usage code'}
reference_document_usage to label (location_in_document)	PATH			document_usage_constraint document_usage_constraint.subject_element_value {document_usage_constraint.subject_element=\ 'location in document'}
reference_document_usage to reference_document (document_referenced)	PATH			document_usage_constraint document_usage_constraint.source-> document <- {document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_relationship.related_document document_relationship {document_relationship.name='document usage reference'} document_relationship.relating_document-> document<- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} {document_product_association.name='equivalence'} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
reference_document_usage to text (usage_description)	PATH			document_usage_constraint document_usage_constraint.subject_element_value {document_usage_constraint.subject_element='usage description'}
RELEASE_AUTHENTICATION	approval	41		
release_authentication to company (release_authority) #1: without person #2: with person	PATH		27	#1:(approval<- approval_person_organization.authorized_approval approval_person_organization {approval_person_organization.role->approval_role approval_role.role='release authentication'} approval_person_organization.person_organization->person_organization_select person_organization_select=organization organization) #2:(approval<- approval_person_organization.authorized_approval approval_person_organization {approval_person_organization.role->approval_role approval_role.role='release authentication'} approval_person_organization.person_organization->person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization->organization)
release_authentication to company_code (release_authority_code)	PATH		28	approval<- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization->person_organization_select=organization organization

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
release_authentication to date (release_date)	PATH		29	approval<- approval_date_time.dated_approval approval_date_time {approval_date_time role_select=approval_date_time role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='release authentication'} approval_date_time.date_time->date_time_select (date_time_select=date) (date_time_select=date_and_time)
release_authentication to identifier (authentication)	PATH			approval approval.status->approval_status approval_status.name
release_authentication to person (person_responsible)	PATH		27	approval<- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization->person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_person->person

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
SECURITY_CLASSIFICATION	security_classification	41		
security_classification to date (classificaiton_date) #1: Date #2: Date and Time #3: Only use this option if a particular instance of security_-classification to be applied to product data has the same classification and declassification dates.	PATH		30	#1: ((security_classification<- security_classification_assignment.assigned_security_classification security_classification_assignment=> applied_security_classification_assignment date_item=applied_security_classification_assignment) #3: (security_classification date_item=security_classification) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date-> date {date_assignment.role-> date_role date_role.name='classification date'}) #2: ((security_classification<- security_classification_assignment.assigned_security_classification security_classification_assignment=> applied_security_classification_assignment date_and_time_item=applied_security_classification_assignment) #3: (security_classification date_and_time_item=security_classification) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date-> date_and_time {date_and_time_assignment.role-> date_time_role date_time_role.name='classification date'})

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
security_classification to date (declassification_date) #1: Date #2: Date and Time	PATH		31	#1: ((security_classification<- security_classification_assignment.assigned_security_classification security_classification_assignment=> applied_security_classification_assignment date_item=applied_security_classification_assignment) (security_classification date_item=security_classification) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment._date-> date {date_assignment.role-> date_role date_role.name='declassification date'}) #2: ((security_classification<- security_classification_assignment.assigned_security_classification security_classification_assignment=> applied_security_classification_assignment date_and_time_item=applied_security_classification_assignment) (security_classification date_and_time=security_classification) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment date_and_time_assignment.assigned_date-> date_and_time {date_and_time_assignment.role-> date_time_role date_time_role.name='declassification date'})

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
security_classification to label (item_classification) #1: Provides alias for security classification level name.	PATH			security_classification {security_classification.purpose='item security classification'} security_classification.security_level-> security_classification_level security_classification_level.name #1: <security_classification_level <identification_item=security_classification_level identification_item<- applied_identificaiton_assignment.item[i] applied_identification_assignment=> identification_assignment {identification_assignment..role-> identification_role identification_role.name=’alias’}>
security_classification to label (title_security_classification) #1: Provide alias for security_-classification_level	PATH			security_classification {security_classification.purpose='title security classification'} security_classification.security_level-> security_classification_level security_classification_level.name #1: <security_classification_level identification_item=security_classification_level identification_item<- applied_identificaiton_assignment.item[i] applied_identification_assignment=> identification_assignment {identification_assignment..role-> identification_role identification_role.name=’alias’}>

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
security_classification to person_- and_organization (classifier) #1: person_and_organization #2: just organization	PATH			#1: (security_classification person_and_organization_item=security_classification person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='classifier officer')}) #2: (security_classification organization_item=security_classification organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='classifier officer')})
SIZE_CHARACTERISTICS	representation	43		{representation.context_of_items-> representation_context (representation_context.context_type='file parameters') (representation_context.context_type='entry parameters') (representation_context.context_type='document parameters')}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
size_characteristics to a_real (byte_size)	PATH			representation representation.items[i]->representation_item {representation_item.name='file size'} representation_item=>measure_representation_item<=measure_with_unit.value_component measure_value measure_value=context_dependent_measure context_dependent_measure {measure_with_unit.unit_component->unit unit=named_unit named_unit=>context_dependent_unit context_dependent_unit.name='byte'}
size_characteristics to label (sheet_size)	PATH			representation {representation.name='document format'} representation.items[i]->representation_item {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=>descriptive_representation_item.description
SIZE_CHARACTERISTICS_-FULL_SIZE	representation	43		({representation.name='one sheet in one file'}) ({representation.name='document size'})
STATUS	action_status	41		
status to identifier (status_code)	PATH			action_status action_status.status

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
status to text (status_code_basis)	PATH			action_status classification_item=action_status classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= {classification_assignment role_select=classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='status code basis'} classification_assignment classification_assignment.assigned_class-> group group.name {group=> class_system}
TDP_ELEMENT_AND_ITEM_- ASSOCIATION	(product_definition_with_associated_- documents) (product_definition_formation_- relationship) (product_definition_relationship) (applied_document_reference)	41 41 41 232		{(product_definition_relationship.name) (product_definition_relationship.name=’fulfills’)}

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
tdp_element_and_item_association to item (related_item)	PATH			(product_definition_with_associated_documents<= prodcuct_definition product_definition.formation-> product_definition_formation) (product_definition_formation_relationship product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) (product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation (applied_document_reference applied_document_reference.items[i]-> document_reference_item document_reference_item=product_definition_formation product_definition_formation)

Table 4 — Mapping table for common (continued)

Application element	AIM element	Source	Rules	Reference path
tdp_element_and_item_association to tdp_element (related_tdp_element)	PATH			(product_definition_with_associated_documents product_definition_with_associated_documents.documentation.ids[i]-> document<- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation prodcut_definition_formation) (product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product-definition_formation) (product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition product_definition.formation-> product_definition_formation) (applied_document_reference<= document_reference document-reference.assigned_document-> document<- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_product_definition.definition_formation product_definition_formation)
TEXT	text	41		
TIME	local_time	41		
time to a_real (second)	local_time.second_component	41		

Table 4 — Mapping table for common (concluded)

Application element	AIM element	Source	Rules	Reference path
time to an-integer (hour)	local_time.hour_component	41		
time to an_integer (minute)	local_time.minute_component	41		
time to text (zone)	PATH			local_time local_time.zone->coordinated_universal_time_offset

Table 5 — Mapping table for data_definition_exchange

Application element	AIM element	Source	Rules	Reference path
ACCESSED_FILE	document_file	232		{document_file<= document document.description= 'access data file'}
accessed_file to text (path_information)	PATH		1	document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')}] [external_identification_assignment.source->external_source external_source.source_id]
accessed_file to text (storage_node_identification)	PATH		1	document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')} identification_assignment.assigned_id
ALTERNATE_-IDENTIFICATION_OF_ITEM	product_definition_formation	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ALTERNATE_ITEM #1: If alternate is a substitute with no concern for which is used in an assembly context #2: If alternate is an alternate item with concern for which is used in an assembly context #3: If alternate is an alternate item with a one for one substitution and allows for two way substitution in a particular assembly context	product	41		#1: ({product<- alternate_product_relationship.alternate}) #2: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute}) #3: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to an_integer (preference_order) #1: (see note 2) #3: (see note 2)	#2: (PATH)			#2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute=> assembly_component_usage_substitute_with_ranking assembly_component_usage_substitute_with_ranking.ranking)
alternate_item to item (interchange_item) #2: (see note 2) #3: (see note 2)	PATH			product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition
alternate_item to item_usage (interchange_item) #1: (see note 2) #3: (see note 2)	PATH			#2: product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- (product_definition_relationship.related_product_definition product_definition_relationship)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to label (type_of_alternate)	PATH			#1: (product<- alternate_product_relationship.alternate alternate_product_relationship alternate_product_relationship.name) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute assembly_component_usage_substitute.name) #3: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product_definition_substitute.name)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to text (usage_conditions) #4:Provides document or portion of document to describe occurrence relationship #5 Multiple usage_conditions can be captured in one text string and parsed, if needed, based on an exchange partner business agreement.	#1: (PATH) #2: (PATH) #3: (PATH)			#1: ({<product<- alternate_product_relationship.basis> <product<- alternate_product_relationship.description> <alternate_product_relationship document_reference_item=alternate_product_relationship document_reference_item<- (applied_document_reference.items[i] applied_document_reference applied_document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document>}) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute #4: <document_reference_item=assembly_component_usage_substitution document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				(applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage->> document_usage_constraint document_usage_constraint.source->) document> assembly_component_usage_substitute.definition) #3: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product_definition_substitute.description)
CONTENT_PROPERTY	representation	43		{representation<- {[representation.name='document content'] [representation.context_of_items->> representation_context representation_context.context_type='document parameters']} property_definition_representation.used_representation property_definition_representation property_definition_representation.defintion->> represented_definition represented_definition=property_definition property_definition {property_definition.name='document property'}}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
content_property to a_real (real_world_scale) #1: real number value #2: text number	PATH			representation representation.items[i]->representation_item=> {representation_item.name='real world scale'} #1: (measure_representation_item<=measure_with_unit [measure_with_unit.value_component->measure_value measure_value=ratio_measure] [measure_withunit.unit_component->unit unit=named_unit named_unit->ratio_unit]) #2: (descriptive_representation_item descriptive_representation_item.description)
content_property to text (detail_level)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='detail level'} descriptive_representation_item descriptive_representation_item.descrption
content_property to text (geometry_type)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='geometry type'} descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
content_property to text (languages) #1: through descrptive text #2: through classification structure	PATH			#1: (representation representation.items[i]-> representation_item=> {representation_item.name='language'} descriptive_representation_item.description) #2: (representation language_item=representation language_item<- language_assignment.items[i] language_assignment<= classification_assignment {classification_assignment.role-> classification_role classification_role.name='language'} classification_assignment.assigned_class-> group {group=> language} group.name)
CONTRACT_SUBMISSION	applied_contract_assignment	232		{applied_contract_assignment<= contract_assignment<- contract_assginment_role.assignment contract_assignment_role contract_assignment_role.role-> contract_role contract_role.name='contract submission'}
contract_submission to contract (delivered_contract)	PATH			applied_contract_assignment<= contract_assignment contract_assignment.assigned_contract-> contract

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to date (date_of_submission) #1: If the date includes date and time #2: If the date only includes a date	PATH		5	applied_contract_assignment #1: (date_and_time_item=applied_contract_assignment date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role-> date_time_role date_time_role.name='date and time of submission'} date_and_time_assignment.assigned_date_and_time-> date_and_time) #2: (date_item=applied_contract_assignment date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment date_assignment.role-> date_role date_role.name='date of submission'} date_assignment.assigned_date-> date)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to text (location) #1: address of just organization #2: address of person in organization	PATH		6	applied_contract_assignment (organization_item=applied_contract_assignment organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='location of contract submission'} organization_assignment.assigned_organization-> organization<- organizational_address.organizations[i] organizational_address<= address) (person_and_organization_item=applied_contract_assignment person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {persion_and_organization_assignment person_and_organization_assignment.role-> persion_and_organization_role person_and_organization_role.name='location of contract submission'} person_and_organization_assignment.assignedperson_and_organization-> person_and_organization persion_and_organization.the_organization-> organization<- organizational_address.organizations[i] organizational_address<= address)
DATA_DEFINITION_ENTRY_-ITEM	(product_definition_with_associated_-documents) (product_definition)	41		{product_definition product_definition.frame_of_reference-> product_definition_context<= application_context_element (application_context_element.name='data definition entry') (application_context_element.name)}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_item to contract_submission (available_from)	PATH			(product_definition_with_associated_documents) (product_definition) contract_item=(product_definition_with_associated_documents)\ (product_- definition) contract_item<- applied_contract_assignment.items[i] applied_contract_assignment
data_definition_entry_item to data_definition_file_entry (entry_files) #1: When file is managed though the item. #2: When the file is just referenced by the item. #3: When the file is the shape of the product. Paths in either #1 or #2 can also be instantiated with Option #3.	PATH			#1: ((product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i]-> document=> document_file) (product_definition document_reference_item=product_definition document_reference_item<- applied_document_reference {applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='mandatory'} applied_document_reference.assigned_document-> document=> document_file)) #2: (product_definition document_reference_item=product_definition document_reference_item<- applied_document_reference {applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				object_role.name='informative'} applied_document_reference.assigned_document-> document=> document_file) #3: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition {property_definition=> product_definition_shape} representation_definition=property_definition representation_definition<- property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit> property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_object characterized_object=> document_file) [(document_file<= document<- product_definition_with_associated_documents.documentation_ids product_definition_with_associated_documents

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				document_reference_item=product_definition_with_associated_documents document_reference_item<- applied_document_reference.assigned_document applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by item') (object_role.name='indentured by item and document')}) (document_file<= document<- applied_document_reference.assigned_document applied_document_reference {applied_document_reference<= document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='informative') (object_role.name='mandatory')} document_reference_itrem=applied_document_reference document_reference_item<- applied_document_reference.assigned_document applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by item') (object_role.name='indentured by item and document')})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_item to delivery_accounting (delivery_accounting_references)	PATH			(product_definition) (product_definition_with_associated_docuemnts<= product_definition) product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='delivery accounting reference'} product_definition_relationship.related_product_definition->product_definition product_definition.formation->product_definition_formation {product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category product_category product_category.name='document'}
data_definition_entry_item to identifier (entry_item_change_level)	PATH			(product_defnition)(product_definition_with_associated_documents) action_item=(product_defnition)\ (product_definition_with_associated_documents) action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name=’change identification for entry’} action_assignment.assigned_action->action {action=> executed_action} action.name

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_item to item_identification (superseded_entry) #1: Identifies superseded item through a definition view relationship #2: Identifies superseded item through a version relationship #3: Identifies superseded item through change request information. The latest item can be identified through its version, view, or structured relationship. The item being superseded can also be identified through its version, view, or structured relationship. #4: Provides approval for superseded request #5: Provides proposed solution to request. #6: Provides status for the request. #7: Provides date and time of the requrest. #8: Provides date and time for when the superseded activity was applied. #9: Provides document or portion of document to describe the superseded change. #10: Provides who requested the change #11: Provides what project this work acitivity is associated with #12: Identifies resourcesto perform the activity. #13: Identifies properties that are being requested.	PATH		7	#1: ((product_definition) (product_definition_with_associated_documents<= product_definition) product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='superseded element'} product_definition_relationship.related_product_definition -> product_definition product_definition.formation-> product_definition_formation) #2: ((product_definition) (product_definition_with_associated_documents<= product_definition) product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship #9: <document_reference_item=product_definition_formation_relationship document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> {product_definition_formation_relationship.name=\ 'superseded version'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #3: ((product_definition action_item=product_definition) (product_definition product_definition.formation-> product_definition_formation

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				action_item=product_definition_formation) (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship action_item=product_definition_relationship) action_item<- applied_action_asignment.items[i] applied_action_assignment<= #8: <applied_action_assignment (date_item=applied_action_assignment date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (date_and_time_item=applied_action_assignment date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> action_assignment action_assignment.assigned_action-> action=> #11: <actioin organizational_project_item=action organizational_project_item<- applied_organizational_project_assignment.items[i] applied_organizational_project_assignment<= organizational_project_assignment organizational_project_assignment.assigned_organizational_project-> organizational_project> #12: <action source_item=action source_item<- action_resource.usage[i] action_resource<- action_resource_relationship.related_resource

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				action_resource_relationship> executed_action=> directed_action directed_action.directive-> action_directive action_directive.requests[i]-> versioned_action_request<- #4:<versioned_action_request approval_item=versioned_action_request approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval> #5: <versioned_action_request<- action_request_solution.request action_request_solution> #6: <versioned_action_request<- action_request_status.assigned_request action_request_status> #7: <(versioned_action_request date_item=versioned_action_requrest date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (versioned_action_request date_and_time_item=versioned_action_requrest date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> #9: <versioned_action_request document_reference_item=versioned_action_request document_reference_item<-

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				(applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document-> document) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment=> document_usage_constraint_assignmen t document_usage_constraint_assignment.assigned_document_usage-> document_usage_constraint)> #10:<versioned_action_request (person_and_organization_item=versioned_action_request person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='requestor')}) (organization_item=versioned_action_request organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='requestor')})> #13 :<versioned_action_request<- action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment applied_aciton_request_assignment.items[i]-> action_request_item action_request_item=property_definition property_definition> action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_aciton_request_assignment.items[i]-> action_request_item (action_request_item=product_definition_formation product_definition_formation) (action_request_item=product_definition product_definition product_definition.formation-> product_definition_formation) (action_request_item=product_definition_relationship product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation))
data_definition_entry_item to notation (entry_notes)	PATH			((product_definition)\ (product_definition_with_associated_documnents) characterized_product_definition=(product_definition)\ (product_definition_with_associated_documnents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_item to reference_document (available_from)	PATH			((product_definition)\ (product_definition_with_associated_documents) document_reference_item=(product_definition)\ (product_definition_with_associated_documents) document_reference_item<- applied_document_referenece.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='source identification'} document_reference.assigned_document-> document<- {docoument.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category product_category.name='document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_item to source_location (available_from)	PATH			product_definition external_identification_item=product_definition external_identification_item<- applied_external_identificatoin_assignmant.items[i] applied_external_identification_assignment
data_definition_entry_item to special_condition (special_conditions)	PATH			(product_defnition)\ (product_definition_with_associated_documents) characterized_product_definition=(product_definition)\ (product_defintion_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item=> descriptive_representation_item
DATA_DEFINITION_ENTRY_-TDP_ELEMENT	(product_definition_with_associated_-documents) (product_definition)	41		{product_definition product_definition.frame_of_reference-> product_definition_context<= application_context_element (application_context_element.name='data definition entry') (application_context_element.name='document version') (application_context_element.name='a representation of a document version') (application_context_element.name)}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to content_property (entry_content_property)	PATH			(product_definition characterized_product_definition=product_definition) (product_definition_with_associated_documents characterized_product_definition=product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name=document content'} {representation.context_of_items-> representation_context representation_context.context_type='document parameters'}
data_definition_entry_tdp_element to contract_submission (available_from)	PATH			(product_definition_with_associated_documents (contract_item=product_definition_with_associated_documents) (product_definition contract_item=product_definition) contract_item<- applied_contract_assignment.items[i] applied_contract_assignment
data_definition_entry_tdp_element to data_definition_file_entry (entry_files)	PATH			product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i]-> document=> document_file

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to delivery_accounting (delivery_accounting_references)	PATH			(product_definition) (product_definition_with_associated_docuemnts<= product_definition) product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='delivery accounting reference'} product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category product_category product_category.name='document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to element_identification (superseded_entry) #1: Identifies superseded element through a definition view relationship #2: Identifies superseded element through a version relationship #3: Identifies superseded element through change request information. The latest element can be identified through its version, view, or structure relationship. The element being superseded can also be identified through its version, view, or structured relationship. #4: Provides approval for superseded request #5: Provides proposed solution to request. #6: Provides status for the request. #7: Provides date and time for the request. #8: Provides date and time for when the superseded activity was applied. #9: Provides document or portion of document to describe the superseded change. #10: Provides who requested the change. #11: Provides what project this work acitivity is associated with #12: Identifies resourcesto perform the activity. #13: Identifies properties that are being requested.	PATH		7	#1: ((product_definition) (product_definition_with_associated_documents<= product_definition) product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='superseded element'} product_definition_relationship.related_product_definition -> product_definition product_definition.formation-> product_definition_formation) #2: ((product_definition) (product_definition_with_associated_documents<= product_definition) product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship #9: <document_reference_item=product_definition_formation_relationship document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> {product_definition_formation_relationship.name='superseded version'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #3: ((product_definition action_item=product_definition) (product_definition product_definition.formation-> product_definition_formation action_item=product_definition_formation)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				(product_definition<- producgt_definition_relationship.related_product_definition product_definition_relationship action_item=product_definition_relationship) action_item<- applied_action_asignment.items[i] applied_action_assignment<= #8: <applied_action_assignment (date_item=applied_action_assignment date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (date_and_time_item=applied_action_assignment date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> action_assignment action_assignment.assigned_action-> action=> #11:<actioin organizational_project_item=action organizational_project_item<- applied_organizational_project_assignment.items[i] applied_organizational_project_assignment<= organizational_project_assignment organizational_project_assignment.assigned_organizational_project-> organizational_project> #12: <action source_item=action source_item<- action_resource.usage[i] action_resource<- action_resource_relationship.related_resource action_resource_relationship>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				executed_action=> directed_action directed_action.directive-> action_directive action_directive.requests[i]-> versioned_action_request<- #4:<versioned_action_request approval_item=versioned_action_request approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval> #5: <versioned_action_request<- action_request_solution.request action_request_solution> #6: <versioned_action_request<- action_request_status.assigned_request action_request_status> #7: <(versioned_action_request date_item=versioned_action_requrest date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (versioned_action_request date_and_time_item=versioned_action_request date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> #9: <versioned_action_request document_reference_item=versioned_action_request document_reference_item<- applied_document_reference.item[i]

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_document_reference=> document_reference document_reference.assigned_document-> document> #10:versioned_action_request (person_and_organization_item=versioned_action_request person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='requestor')}) (organization_item=versioned_action_request organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='requestor')})> #13: <versioned_action_request<- action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment applied_aciton_request_assignment.items[i]-> action_request_item action_request_item=property_definition property_definition> action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment applied_aciton_request_assignment.items[i]-> action_request_item (action_request_item=product_definition_formation product_definition_formation) (action_request_item=product_definition product_definition

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition.formation->product_definition_formation) (action_request_item=product_definition_relationship product_definition_relationship product_definition_relationship.related_product_definition->product_definition product_definition.formation->product_definition_formation))
data_definition_entry_tdp_element to file_format (entry_format)	PATH		16	((product_definition)\ (product_definition_with_associated_documents) characterized_product_definition=(product_definition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='document format'})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to identifier (entry_item_change_level)	PATH			(product_defnition)(product_definition_with_associated_documents) action_item=(product_defnition)\ (product_definition_with_associated_documents) action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification for entry'} action_assignment.assigned_action-> action {action=> executed_action} action.name

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to label (actual_inclusion_in_data_-exchange)	PATH			(product_definition_with_associated_documents)\ (product_defintion) characterized_product_definition= (product_definition_with_associated_-documents)\ (product_defintion) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='include flag'} representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} representation_item.name {(descriptive_representation_item.name) (descriptive_representation_item.name='already delivered') (descriptive_representation_item.name='available for access') (descriptive_representation_item.name='fully included') (descriptive_representation_item.name='not deliverable') (descriptive_representation_item.name='not delivered yet') (descriptive_representation_item.name='not received') (descriptive_representation_item.name='partially included') (descriptive_representation_item.name='receipt accepted') (descriptive_representation_item.name='receipt acknowledged') (descriptive_representation_item.name='receipt partially rejected') (descriptive_representation_item.name='receipt rejected') (descriptive_representation_item.name='resubmission')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to label (data_usage_rights)	PATH			((product_defnition)(product_definition_formation) characterized_product_definition=(product_defnition)\ (product_definition_formation) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} approval_item=property_definition approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='data usage rights'} approval_assignment.assigned_approval-> approval approval.level <approval.status-> approval_status>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to notation (entry_notes)	PATH			((product_definition)\ (product_definition_with_associated_documnents) characterized_product_definition=(product_definition)\ (product_definition_with_associated_documnents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to reference_document (available_from)	PATH			(product_definition)\ (product_definition_with_associated_documents) document_reference_item=(product_definition)\ (product_definition_with_associated_documents) document_reference_item<- applied_document_referenece.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='source identification'} document_reference.assigned_document-> document<- {docoument.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category product_category.name='document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to size_characteristics (size)	PATH			((product_defnition)\ (product_definition_with_associated_documents) characterized_product_definition=(product_defnition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document size'})
data_definition_entry_tdp_element to source_location (available_from)	PATH			product_definition external_identification_item=product_definition external_identification_item<- applied_external_identificatoin_assignmant.items[i] applied_external_identification_assignment

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to special_condition (special_conditions)	PATH			(product_defnition)\ (product_definition_with_associated_documents) characterized_product_definition=(product_definition)\ (product_defintion_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to text (entry_format)	PATH			(product_defnition)\ (product_definition_with_associated_documents) characterized_product_definition=(product_defnition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> representation_item=> descriptive_representation_item {(descriptive_representation_item.name) (descriptive_representatoin_item.name='size format') (descriptive_representatoin_item.name='character code') (descriptive_representatoin_item.name='data format')} descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_entry_tdp_element to yes_no (master_file)	PATH			(product_definition_with_associated_documents characterized_product_definition=product_definition_with_associated_documents) (product_definition characterized_product_definition=product_definition) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='master representation flag'} representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} {(representation_item.name='master representation') (representation_item.name='not master representation')}
DATA_DEFINITION_- EXCHANGE	product_definition_formation	41	8 24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='data definition exchange'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange to data_-definition_exchange_body (list_body)	PATH			product_definition_formation<- product_definition.formation (product_defnition)(product_definition_with_associated_documents) {product_definition.description='data definition exchange content'}
data_definition_exchange to data_-definition_exchange_header (list_header)	PATH			product_definition_formation<- product_definition.formation (product_defnition)(product_definition_with_associated_documents) {product_definition.description='data definition exchange content'} characterized_product_definition=(product_defnition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='data definition exchange header'}
data_definition_exchange to list_-presentation (presentation)	PATH		9	product_defnition_formation presented_item_item=product_defnition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation.presentation->presentation_representation_select presentation_representation_select=presentation_set presentation_set
DATA_DEFINITION_-EXCHANGE_BODY	product_definition	41		{product_definition product_definition.description= 'data definition exchange content'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_body to data_definition_indentured_list_-method (list_method)	PATH		21	product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association=> document_product_equivalence<= document_product_association document_product_association.relating_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by document') (object_role.name='indentured by item') (object_role.name= indentured by item and document')} applied_document_reference

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_body to notation (notes_list)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='data definition exchange body property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name="notation"} representation.items[i]-> representation_item=> descriptive_representation_item

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_body to revision (revision_history)	PATH			product_definition <product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action=> {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_body to simple_list_of_elements (list_method)	PATH			product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association=> document_product_equivalence<= document_product_association document_product_association.relating_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='list of elements'} applied_document_reference

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_body to simple_list_of_files (list_method)	PATH			product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association=> document_product_equivalence<= document_product_association document_product_association.relating_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='list of files'} applied_document_reference
DATA_DEFINITION_-EXCHANGE_HEADER	property_definition	41		{property_definition property_definition.name='data definition exchange header'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_header to date (date_of_transfer) #1: just date #2: date and time #3: date through product view #4: date through product version	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition #1: (#3: (date_item=product_definition) #4: (product_definition.formation-> product_definition_formation date_item=product_definition_formation) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role-> date_role date_role.name='transfer date'} date_assignment.assigned_date-> date) #2: (#3: (date_and_time_item=product_definition) #4: (product_definition.formation-> product_definition_formation date_and_time_item=product_definition_formation) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role-> date_time_role date_time_role.name='transfer date'} date_and_time_assignment.assigned_date_and_time-> date_and_time date_and_time.date_component-> date)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_header to delivery_accounting (delivery_accounting_references) #1: reference full document #2: reference portion of document	PATH			property_definition document_reference_item=property_definition document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='delivery accounting'} document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document <- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_header to element_identification (procurement_references)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.relating_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'procurement reference'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation
data_definition_exchange_header to header (common_header)	IDENTICAL_MAPPING			
data_definition_exchange_header to item_identification (procurement_references)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.relating_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'procurement reference'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation
data_definition_exchange_header to reason (exchange_reason)	IDENTICAL MAPPING			

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_header to system_destination (destinations)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition product_defnition.formation-> product_definition_formation (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role person_and_organization_role.name='system destination'} person_and_organization_assignment.assigned_person_and_organization-> person_and_organization) (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role organization_role.name='system destination'} organization_assignment.assigned_organization-> organization)
DATA_DEFINITION_-EXCHANGE_SIMPLE_ENTRY	product_definition_formation	41		{product_definition_formation product_definition_formatin.of_product-> product<- product_related_product_category.product[i] product_related_product_categoy<= product_category product_category.name='document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_exchange_simple_-entry to data_definition_entry_-tdp_element (entry)	PATH			product_definition_formation<-product_defintion.formation product_definition
data_definition_exchange_simple_-entry to element_identification (simple_entry)	IDENTICIAL MAPPING			
DATA_DEFINITION_-INDENTURED_ENTRY #1: When entry is a document #2: When entry is an item	#1,#2:(product_definition) (product_-definition_with_associated_documents)	41 41		#1,#2:{ (product_definition)\ (product_definition_with_associated_documents)}
data_definition_indentured_entry to data_definition_entry_item (entry_characteristics) #1: (see note 2)	#2: (IDENTICAL MAPPING)			
data_definition_indentured_entry to data_definition_entry_tdp_element (entry_characteristics) #2: (see note 2)	#1: (IDENTICAL MAPPING)			

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_entry to effectivity (effective_on) #3: Provide effectivity based on usage #4: Provide effectivity for view or instance. #5: Provide effecitivity for version #6: Provide effecitivity based on item instance or relationship	PATH			#3:(product_definition<- product_definition_relationship.related_product_definition product_definition_relationship<- product_definition_effectivity.usage product_definition_effectivity=> configuration_effectivity) (#4::(prodcut_definition effectivity_item=product_definition) #5: (product_definition product_definition.formation-> product_definition_formation effectivity_item=product_definition_formation) #6: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship effectivity_item=product_definition_formation) effectivity_item<- applied_effectivity_assignment.item[i] applied_effectivity_assignment=> effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity)
data_definition_indentured_entry to item_parent_to_item_child_-relationship (indentured_entry) #3: Allows indentured branching from an alternate part level and part occurrence level.	PATH			product_definition<- product_definition_relationship.related_product_definition (product_definition_relationship) (product_definition_relationship=> (assembly_component_usage)\ (assembly_component_usage_substitute)\ (next_assembly_usage_occurrence)\ (quantified_assembly_component_usage)\ (assembly_component_usage_with_ranking)\ (make_from_usage_option_with_reference_designator)\ (make_from_usage_option)) #3: ((alternate_product_relationship) (product_definition_occurrence_relationship))

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_entry to item_parent_to_tdp_element_-child_relationship (indentured_entry)	PATH			((product_definition)(product_definition_with_associated_documents) product_definition_relationship.related_product_definition product_definition_relationship) ((product_definition)(product_definition_with_associated_documents) document_reference_item=(product_definition)\ (product_definition_with_associated_documents) document_reference_item<- applied_document_reference.items[i] applied_document_reference)
data_definition_indentured_entry to label (indenture_level)			20	product_definition characterized_product_definition=product_definition characterized_defnition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='indentured level tag'} representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} representation_item.name
data_definition_indentured_entry to tdp_element_parent_to_item_-child_relationship (indentured_entry)	PATH			(product_definition)(product_definition_with_associated_documents)<- product_definition_relationship.related_product_definition product_definition_relationship

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_entry to tdp_element_parent_to_tdp_-element_child_relationship (indentured_entry)	PATH			(product_definition)(product_definition_with_associated_documents)<- product_definition_relationship.related_product_definition product_definition_relationship
data_definition_indentured_entry to tdp_indentured_item (indentured_entry) #1: (see note 2)	#2: (IDENTICAL MAPPING)			
data_definition_indentured_entry to tdp_indentured_tdp_element (indentured_entry) #2: (see note 2)	#1: (IDENTICAL MAPPING)			
DATA_DEFINITION_-INDENTURED_LIST_METHOD	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by document') (object_role.name='indentured by item') (object_role.name= indentured by item and document')}
data_definition_indentured_list_-method to accessed_file (order_of)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item document_reference_item=document_file document_file

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_list_- method to exchange_file (order_of)				applied_document_reference applied_document_reference.items[i]->document_reference_item document_reference_item=document_file document_file

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_list_- method to data_definition_- indentured_entry (order_of) #1: point through arc to a subordinate entry #2: point through arc to a parent entry #3: point directly to top entry	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item #1: <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)> #1: <document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.related_product_definition-> product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)> #2: <document_reference_item=applied_document_reference applied_document_reference<= document_reference document_reference.assigned_document-> document<- document_product_association.relating_document document_product_association=> document_product_equivalence<= document_product_association document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=(product_definition_formation product_definition_formation<- product_definition.formation product_definition) (product_definition)> #2: <document_reference_item=product_definition_with_associated_documents product_definition_with_associated_documents> #2: <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation->

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)> #2: < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)> #3: <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)> #3: <document_reference_item=product_definition> #1: <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents))>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
data_definition_indentured_list_- method to indentured_list_by_- document (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='indentured by document'}
data_definition_indentured_list_- method to indentured_list_by_part (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='indentured by item'}
data_definition_indentured_list_- method to indentured_list_by_- part_with_document_references_- to_parts (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='indentured by item and document'}
DATE_EFFECTIVITY	dated_effectivity	41		
date_effectivity to date (end_date)	PATH			dated_effectivity dated_effectivity.effectivity_end_date->date_and_time

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
date_effectivity to date (start_date)	PATH			dated_effectivity dated_effectivity.effectivity_start_date-> date_and_time
DELIVERY_ACCOUNTING	product_definition_formation	41		
delivery_accounting to identifier (reference_identification)	PATH			product_definition_formation [product_definition_formation.id] [product_defiition_formation.of_product-> product product.id]
delivery_accounting to text (reference_identification_-description)	PATH			product_definition_formation product_definition_formation.of_product-> product product.description
DOCUMENT_USAGE_-PARAMETER	document_usage_constraint	41		
document_usage_parameter to label (subject)	PATH			document_usage_constraint document_usage_constraint.subject_element_value {(document_usage_constraint.subject _element_value) (document_usage_constraint.subject_element_value='type') (document_usage_constraint.subject_element_value='grade') (document_usage_constraint.subject_element_value='form') (document_usage_constraint.subject_element_value='composition')}
document_usage_parameter to text (value_of)	PATH			document_usage_constraint document_usage_constraint.subject_element_value
DRAWING_SUFFIX_NUMBER_-COMBINATION	product	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_suffix_number_- combination to element_- identification (drawing_number)	PATH	41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=\ 'drawing suffix number combination'} product_relationship.related_product->product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category.name='drawing'}
drawing_suffix_number_- combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name=\ 'drawing suffix number combination'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
EFFECTIVITY #1: When the effectivity is for a part and the next higher part is known #2: When the effectivity is for a part and the next higher part is not known	#1: (<configuration_effectivity> <(serial_numbered_effectivity) (dated_effectivity) (lot_effectivity) (time_interval_based_effectivity)> #2:[applied_effectivity_assignment] [applied_effectivity_context_assignment]	41 41 41 41 41 232 232		#1: ({configuration_effectivity<= product_definition_effectivity product_definition_effectivity.usage->product_definition_relationship product_definition_relationship.description='item usage'}) #2: ({applied_effectivity_assignment<= effectivity_assignment<- effectivity_context_assignment.assigned_effectivity_assignment effectivity_context_assignment [effectivity_context_assignment.role->effectivity_context_role (effectivity_context_role.name) (effectivity_context_role.name='context document of effectivity') (effectivity_context_role.name='context organization of effectivity')] [effectivity_context_assignment=> applied_effectivity_context_assignment applied_effectivity_context_assignment.items[i]->effectivity_context_item (effectivity_context_item=organization) (effectivity_context_item=product_definition_formation product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<= product.category product_category.name='document')]})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to approval (effectivity_approvals)	PATH			#1: (configuration_effectivity approval_item=configuration_effectivity approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) #2: (applied_effectivity_assignment approval_item=applied_effectivity_assignment approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role>
effectivity to label (name)	PATH			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.named_item name_attribute name_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.described_item name_attribute name_attribute.attribute_value)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to label (work_activity)	PATH			configuration_effectivity action_item=configuration_effectivity action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.name
effectivity to product_configuration (product)	PATH			#1: (configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity=> product_definition_effectivity=> configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to text (description)	PATH			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attrubute.described_item description_attribute description_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attrubute.described_item description_attribute description_attribute.attribute_value)
EVENT	event_occurrence	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
event to date (actual_date) #1: Just date #2: date and time	PATH			event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name='actual date'} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name='actual date'} date_assignment.assigned_date_and_time-> date_and_time)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
event to date (planned_date)				event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name=’planned date’} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name=’planned date’} date_assignment.assigned_date_and_time-> date_and_time)
event to identifier (id)	event_occurrence.id	41		
event to label (name)	event_occurrence.name	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
event to person_and_organization (responsible_person_organization)	PATH			event_occurrence (person_and_organization_item=event_occurrence person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role person_and_organization_role.name='identifier'} person_and_organization_assignment.assigned_person_and_organization-> person_and_organization) (organization_item=event_occurrence organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role organization_role.name='system destination'} organization_assignment.assigned_organization-> organization)
event to project (assignment)	PATH			event_occurrenc<- event_occurrence_assignment.assigned_event_occurrence event_occurrence_assignment<= applied_event_occurrence_assignment applied_event_occurrence_assignment.items[i]-> event_occurrence_item event_occurrence_item=organizational_project organizational_project
event to text (description)	event_occurrence.description	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
EXCHANGE_FILE	document_file	232		{document_file<= document document.description='exchange data file'}
exchange_file to label (included_in_exchange)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='exchange entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='include flag'} representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} representation_item.name {(representation_item.name) (representation_item.name='already delivered') (representation_item.name='available for access') (representation_item.name='fully included') (representation_item.name='not deliverable') (representation_item.name='not delivered yet') (representation_item.name='not received') (representation_item.name='partially included') (representation_item.name='receipt accepted') (representation_item.name='receipt acknowledged') (representation_item.name='receipt partially rejected') (representation_item.name='receipt rejected') (representation_item.name='resubmission')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
exchange_file to system_- declaration (destination_system)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='destination system declaration'}
EXTERNAL_LIBRARY_- REFERENCE #1: If external_library_reference is an element in a class source select, that is Reference by a general_-classification as the classification_-source. #2: If external_library_reference is an element in a property_source _-seleect, that is reference by a property as the property_source.	#1: (externally_defined_class) #2: (externally_defined_general_-property)	232 232		#1: (externally_defined_class<= [ckass<= group][externally_defined_item]) #2:(externally_defined_general_property<= [general_property] [externally_defined_item])
external_library_reference to identifier (external_id)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.item_id

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
external_library_reference to identifier (library_type)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source-> external_sourrce external_sourrce.id {external_source.source_id-> source_item source_item=identifier identifier
external_library_reference to text (description)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.cource-> exterrnal_source external_sourcee.description
FILE	document_file	232		{document_file<= document<- document_representation_type.representation_types document_representation_type (document_representation_type.name='digital') (document_representation_type.name='physical')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to change_identification (change_status)	PATH			document_file action_item=document_file action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> executed_action
file to content_property (file_content_property)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name=document content'} {representation.context_of_items-> representation_context representation_context.context_type='document parameters'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to distribution_notice (distribution_authorizations)	PATH			document_file approval_item=document_file approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name=distribution notice'} approval_assignment.assigned_approval-> approval
file to file_format (context_file_format)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to identifier (context_file_name) #1: provides alias for file identification	PATH			document_file<= document document.id #1<document _file identification_item=document_file identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}>
file to identifier (native_format_file_name)	PATH			(document_file<= document<- document_relationship.related_document document_relationship {document_relationship.name='translation'} document_relationship.relating_document-> document {document=> document_file} document.id
file to label (file_type_content)	PATH			document_file<= document document.kind-> document_type document_type.product_data_type

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to notation (file_note)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='notation'} representation.items[i]->representation_item=> descriptive_representation_item
file to person_organization (assigned_to) #1 person and orgnaization #2: just organization	PATH			#1: (document_file persion_and_organization_item=document_file person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization->person_and_organization person_organization_select=persion_and_organization person_organization_select) #2: (document_file organization_item=document_file organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization->organization person_organization_select=organization person_organization_select)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to security_classification (security_identification)	PATH			document_file security_classification_item=document_file security_classification_item<- applied_security_classification_assignment.items[i] applied_security_classification_assignment<= security_classification_assignment <security_classification_assignment role_select=security_classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role> security_classification_assignment.assigned_security_classification-> security_classification
file to size_characteristics (size)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file to system_declaration (source_system)	PATH			document_file characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {(representation.name='source system declaration') (representation.name='document creation')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_FORMAT	representation	43		{representation.name='document format'}
file_format to date (release_date)	PATH			representation (date_item=representation date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date-> date {date_assignment.role-> date_role date_role.name='release date'}) (date_and_time_item=representation date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role-> date_time_role date_time_role.name='release date'} date_and_time_assignment.assigned_date_and_time-> date_and_time)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (change_level)	PATH			representation action_item=representation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action {action=> executed_action} action.name {action.chosen_method-> action_method action_method.name='change level'}
file_format to identifier (format_code)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='character code'} descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (revision_level)	PATH			representation action_item=representation acton_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action action.name {action=> executed_action) {action.chosen_method-> action_method action_method.name='revision level'}
file_format to text (format_standard)	PATH			representation {representation.name='document format'} representation.items[i]-> representation_item=> {representation_item.name='data format'} descriptive_representation_item descriptive_representation_item.description
FILE_RELATIONSHIP	document_relationship	41		
file_relationship to file (related_file)	PATH			document_relationship document_relationship.related_document-> document=> document_file

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
file_relationship to file (relating_file)	PATH			document_relationship document_relationship.related_document-> document=> document_file
file_relationship to label (name)	PATH			document_relationship document_relationship.name
file_relationship to text (description)	PATH			document_relationship document_relationship.description
INDENTURED_LIST_BY_-DOCUMENT	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_document to tdp_indentured_tdp_element (top_indentured_tdp_elements)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition>
INDENTURED_LIST_BY_PART	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part to tdp_- indentured_item (top_indentured_items)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents))>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
INDENTURED_LIST_BY_- PART_WITH_DOCUMENT_- REFERENCES_TO_PARTS	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item and document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part_with_- document_references_to_parts to tdp_indentured_item (top_indentured_item)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <(document_reference_item=product product<- product_definition_formation.of_product) (document_reference_item=product_definition_formation) product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents))>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part_with_- document_references_to_parts to tdp_indentured_tdp_element (top_indentured_item)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition (product_definition) (product_definition=> product_definition_with_associated_documents)) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition (product_definition) (product_definition=> product_definition_with_associated_documents))>
INDEPENDENT_PROPERTY	general_property	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property to external_-library_reference (property_source)	IDENTICAL MAPPING			{general_property=> externally_defined_general_property}
independent_property to identifier (id)	general_property.id	41		
independent_property to measure (allowed_unit) #1: If the unit has a commonly used descriptor. #2: If the unit descriptor is constructed of more than one unit name.	PATH			general_property represented_definition=general_property represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='allowed units'} representation.context_of_item-> representation_context=> global_unit_assigned_context.units[i]-> unit #1: (unit=named_unit named_unit) #2: (unit=derived_unit derived_unit)
independent_property_to text (description)	general_property.description	41		
INDEPENDENT_PROPERTY_-RELATIONSHIP	general_property_relationship	41		
independent_property_relationship to text (description)	general_property_relationship.description	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property_relationship to independent_property (related_property)	PATH			general_property_relationship general_property_relationship.related_property->general_property
independent_property_relationship to independent_property (relating_property)	PATH			general_property_relationship general_property_relationship.relating_property->general_property
independent_property_relationship to label (relation_type)	general_property_relationship.name	41		
INDEPENDENT_PROPERTY_-USAGE	general_property_association	41		
independent_property_usage to independent_property (assigned_property)	PATH			general_property_association general_property_association.base_definition->general_property
independent_property_usage to property_assignment (assigned_to)	PATH			general_property_association general_property_association.derived_definition->derived_property_select derived_property_select=property_definition property_definition
ITEM_IDENTIFICATION	product_definition_formation	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification.. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3:(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #2:(product_definition_formation-> product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product-> product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification-> certification

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to change_- identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_-authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
ITEM_PARENT_TO_ITEM_- CHILD_RELATIONSHIP	product_definition_relationship	41		
item_parent_to_item_child_- relationship to item_identification (child)	PATH			product_definition_relationship product_definition_relationship.related_product_definition->(product_definition)(product_definition_with_associated_documents) product_definition.formation->product_definition_formation
item_parent_to_item_child_- relationship to item_identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition->(product_definition)(product_definition_with_associated_documents) product_definition.formation->product_definition_formation
ITEM_PARENT_TO_TDP_- ELEMENT_CHILD_- RELATIONSHIP	(product_definition_relationship) (applied_document_reference)	41 232		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_parent_to_tdp_element_- child_relationship to element_- identification (child)	PATH			(product_definition_relationship product_definition_relationship.related_product_definition-> (product_definition)(product_definition_with_associated_documents) product_definition.formation-> product_definition_formation) (applied_document_reference<= document_reference document_reference.assigned_document-> document<- {document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_definition product_or_formation_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category product_category.name='document'})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
item_parent_to_tdp_element_- child_relationship to item_- identification (parent)	PATH			(product_definition_relationship product_definition_relationship.relating_product_definition->(product_definition)(product_definition_with_associated_documents) product_definition.formation->product_definition_formation) (applied_document_reference applied_document_reference.items[i]->document_reference_item document_reference_item=product_definition_formation product_definition_formation)
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role->organization_role organization_role.name='code administator'} organization_assignment.assigned_organization->organization
item_type to label (item_code)	PATH			product_category product_category.name
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description
LOT_EFFECTIVITY	lot_effectivity	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
lot_effectivity to a_number (lot_size)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.value_component-> measure_value
lot_effectivity to identifier (lot_identification)	PATH			lot_effectivity lot_effectivity.effectivity_lot_id
lot_effectivity to measure (lot_unit_of_measure)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.unit_component-> unit
MEASURE	named_unit	41		
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.items[i] representation representation.name='notation'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #!1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role->classification_role classification_role.name='reference code'} classification_assignment.assigned_class->group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_role {classification_assignment classification_assignment.role->classification_role classification_role.name='type of notation'} classification_assignment.assigned_class->group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
NUMBER_WITH_UNITS	measure_with_unit	41		
number_with_units to a_real (value_of)	PATH			measure_with_unit mesure_with_unit.value_component measure_value
number_with_units to measure (units_of)	PATH			measure_with_unit measure_with_unit.unit_component unit
PROJECT	organizational_project	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
project to date (end_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role->date_role (date_role.name) (date_role.name=’actual end’) (date_role.name='planned end')} date_assignment.assigned_date->date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role->date_time_role (date_time_role.name) (date_time_role.name=’actual end’) (date_time_role.name='planned end')} date_and_time_assignment.assigned_date->date_and_time)

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
project to date (start_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role-> date_role (date_role.name) (date_role.name='actual start') (date_role.name='planned start')} date_assignment.assigned_date-> date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role-> date_time_role (date_time_role.name) (date_time_role.name='actual start') (date_time_role.name='planned start')} date_and_time_assignment.assigned_date-> date_and_time)
project to identifier (id)	organizational_project.id	41		organizational_project id_attribute_select=organizational_project id_attribute_select<- id_attribute.identified_item id_attribute id_attribute.attribute_value
project to label (name)	organizational_project.name	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
project to perrson_and_- organization (participants)	PATH			organizational_project (person_and_organization_item=organizational_project person_and_organization_item<- applied_person_and_organization_assignment.item[i] applied_person_and_organization_assignment=> person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=person_and_organization person_organization_select) (organization_item=organizational_project organization_item<- applied_organization_assignment.item[i] applied_organization_assignment=> organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
project to person_and_organization (project_owner)	PATH			organizational_project organizational_project.responsible_organization[i]-> organization person_organization_select=organization person_organization_select
project to text (description)	organizational_project.description	41		
PROPERTY_ASSIGNMENT	property_definition	41		
property_assignment to identifier (described_element)	PATH			property_definition property_definition.definition-> characterized_definition

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
property_assignment to label (name)	property_definition.name	41		
property_assignment to text (description)	property_definition.description	41		
QUANTITY	measure_with_unit	41		
quantity to a_number (value_of)	PATH			measure_with_unit measure_with_unit.value_component-> measure_value measure_value=count_measure count_measure
quantity to text (quantity_accuracy)	PATH			measure_with_unit<- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifier-> type_qualifier type_qualifier.name {(type_qualifier.name) (type_qualifier.name='exact') (type_qualifier.name='as required') (type_qualifier.name='approximate')}
quantity to text (value_of)	PATH			measure_with_unit measure_with_unit.value_component-> measure_value measure_value=descriptive_measure descriptive_measure
quantity to volume (value_of)	IDENTICAL MAPPING			

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
quantity to weight (value_of)	PATH			measure_with_unit=> measure_representation_item<= representation_item
REASON	property_definition	41		
reason to exchange_reason (base)	PATH		14	property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_represenation property_definition_represenation.used_representation-> representation {representation.name='exchange reason'} representation.items[i]-> represenation_item=> {representation_item.name='base exchange reason'} descriptive_representation_item {(descriptive_representation_item.description) (descriptive_representation_item.description='acknowledge receipt of delivery') (descriptive_representation_item.description='engineering design analysis') (descriptive_representation_item.description='interim engineering design review') (descriptive_representation_item.description='final design review') (descriptive_representation_item.description='initial data submittal') (descriptive_representation_item.description='procurement design package') (descriptive_representation_item.description='provisioning data submittal') (descriptive_representation_item.description='request for proposal') (descriptive_representation_item.description='revision update data submittal') (descriptive_representation_item.description='specification design package')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
reason to item_identification (applicable_to)	PATH			property_definition property_definition.definition->characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition<-{product_definition.description='data definition exchange content'} product_definition_relationship.related_product_definition product_definition_relationship{product_definition_relationship.name='applicable to exchange reason'} product_definition_relationship.relating_product_definition->product_definition->product_definition.formation
reason to text (exchange_purpose)	PATH		14	property_definition represented_definition=property_definition represented_definition<-property_definition_representation.definition property_definition_represenation property_definition_represenation.used_representation->representation{representation.name='exchange reason'} representation.items[i]->represenation_item=>{representation_item.name='exchange purpose'} descriptive_representation_item descriptive_representation_item.description
RELATIVE_EVENT	relative_event_occurrence	41		
relative_event to event (related_event)	PATH			relative_event_occurrence relative_event_occurrence.base_event->event_occurrence

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
relative_event to number_with_- units (offset)	PATH			relative_event_occurrence relative_event_occurrence.offset-> time_measure_with_unit=> measure_with_units
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly to Revision when approval of person or organization is not needed. #2: date and time associated directly to Revision when approval of person or organization is not needed. #3 When approval of person or organization is needed and the date of their approval	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
revision to other_reference_- document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
revision to person_and_- organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
SEQUENCE_EFFECTIVITY	serial_numbered_effectivity	41		
sequence_effectivity to identifier (from_effectivity_id)	PATH			serial_numbered_effectivity serial_numbered_effectivity.effectivity_start_id
sequence_effectivity to identifier (thru_effectivity_id)	PATH			serial_numbered_effectiivity serial_numbered_effectivity..effectivity_end_id
sequence_effectivity to measure (quantity_unit_of_measure)	PATH		10	serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit measure_with_unit.unit_component-> unit
sequence_effectivity to quantity (total_component_quantity_in_-product_configuration)	PATH			serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {prodcut_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
SIMPLE_LIST_OF_ELEMENTS	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='list of elements'}
simple_list_of_elements to data_-definition_exchange_simple_entry (element_entries)	PATH		32	applied_document_reference applied_document_reference_.items[i]-> document_reference_item document_reference_item=product_definition_formation product_definition_formation
SIMPLE_LIST_OF_FILES	applied_document_reference	232		{applied_document_reference<= {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='list of files'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
simple_list_of_files to data_-definition_file_entry (file_entries)	PATH		33	applied_document_reference applied_document_reference.items[i]->document_reference_item document_reference_item=document_file document_file {document_file<= document<- document_representation_type.representation_types document_representation_type (document_representation_type.name='digital') document_representation_type.name='physical')}
SIZE_CHARACTERISTICS_-INTERNAL_DIVISIONS	representation	43		{representation.name='multiple sheets in one file'}
size_characteristics_internal_-division to an_integer (internal_division_count)	PATH			representation representation.items[i]->representation_item {representation_item.name='internal division count'} representation_item=> measure_representation_item
size_characteristics_internal_-division to label (internal_-division_type)	PATH			representation representation.items[i]->representation_item {representation_item.name='internal division type'} representation_item representation_item=> descriptive_representation_item.description
SIZE_CHARACTERISTICS_-SHEET_ACROSS_FILE	representation	43		{representation.name='file containing portion of sheet'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
size_characteristics_sheet_across_-file to an_integer (frame_number_-comprising_this_file)	PATH			representation representation.items[i]->representation_item {representation_item.name='sheet portion number'} representation_item=>measure_representation_item
size_characteristics_sheet_across_-file to an_integer (number_of_frames_for_sheet)	PATH			representation representation.items[i]->representation_item {representation_item.name='number of portions sheet is divided into'} representation_item=>measure_representation_item
SOURCE_INFORMATION_TYPE	representation	43	22	{representation representation.name='source information type'}
source_information_type to identifier (source_code)	PATH			representation representation.items[i]->representation_item {representation_item=>descriptive_representation_item} representation_item.name
source_information_type to text (source_description)	PATH			representation representation.items[i]->representation_item=>descriptive_representation_item descriptive_representation_item.description
source_information_type to text (type_of_coding_scheme)	PATH			representation representation.context_of_items[i]->representation_context representation_context.context_type

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
SOURCE_LOCATION	applied_external_identification_-assignment	232		
source_location to text (path_information)	PATH			applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role-> (identification_role) (identification_role.description='access context')}] [external_identification_assignment.source-> external_source external_source.source_id]
source_location to text (storage_node_identification)	PATH			applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role-> (identification_role) (identification_role.description='access context')} identification_assignment.assigned_id
SPECIAL_CONDITION	descriptive_representation_item	45		
special_condition to label (code)	PATH			descriptive_representation_item<= representation_item representation_item.name
special_condition to text (description)	PATH			descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
special_condition to text (type_of_coding_scheme)	PATH			descriptive_representation_item<= representation_item<- representation.items[i] representation {representation.name='special condition'} representation.context_of_items[i]-> representation_context representation_context.context_type
SPECIFICATION_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] prodcut_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference_document'} product_category_relationship.category product_category_relationship product_category-relationsip.sub_category-> product_category product_category.name='specification document'}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
STANDARD_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] prodcut_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference_document'} product_category_relationship.category product_category_relationship product_category-relationsip.sub_category-> product_category product_category.name='standard document'}
SYSTEM_DECLARATION	representation	43		{(representation.name='source system declaration') (representation.name='destination system declaration') (representation.name='document creation')}

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to identifier (system) #1: Use when system operating system, creating_interface, and system_related_element_identifier are needed. #2 : Use when system operating system, creating_interface, and system_related_element_identifier are not needed.	PATH			#1: (representation representation.items[i]->representation_item=> {(representation_item.name='computer system') (representation_item.name='creating system')} descriptive_representation_item descriptive_representation_item.description) #2: (document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role->identification_role (identification_role.description) (identification_role.description='source system') (identification_role.description='destination system')}] [external_identification_assignment.source->external_source external_source.source_id])

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to identifier (system_element_identifier) #1: Use when system operating system, creating_interface, and system_related_element_identifier are needed. #2 : Use when system operating system, creating_interface, and system_related_element_identifier are not needed.	PATH			#1: (representation representation.items[i]->representation_item=> {representation_item.name='element on computer system'} descriptive_representation_item descriptive_representation_item.description) #2: (document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')} identification_assignment.assigned_id)
system_declaration to identifier (system_related_element_identifier)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='related element on computer system'} descriptive_representation_item descriptive_representation_item.description
system_declaration to text (creating_interface)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='creating interface'} descriptive_representation_item descriptive_representation_item.description

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to text (operating system)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='operating system'} descriptive_representation_item descriptive_representation_item.description
SYSTEM_DESTINATION #1: With Person #2: Without Person	#1: person_and_organization #2: organization	41 41		
system_distination to company (target_company)	#1: (PATH) #2: (IDENTICAL MAPPING)			#1:(person_and_organization person_and_organization.the_organization->organization)
system_destination to company_-code (target_company_code)	#1: (PATH) #2: (PATH)			#1:(person_and_organization person_and_organization.the_organization->organization) #2: (organization)
system_destination to person (target_person) #2: (see note 2)	#1: (PATH)			#1:(person_and_organization person_and_organization.the_person->person)
TDP_ELEMENT	product_definition_formation	41		
TDP_ELEMENT_PARENT_TO_-ITEM_CHILD_RELATIONSHIP	product_definition_relationship	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
tdp_element_parent_to_item_- child_relationship to element_- identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition->product_definition product_definition.formation->product_definition_formation {product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product_category product_category.name='document'}
tdp_element_parent_to_item_- child_relationship to item_- identification (child)	PATH			product_definition_relationship product_definition_relationship.related_product_definition ->product_definition product_definition.formation->product_definition_formation
TDP_ELEMENT_PARENT_TO_- TDP_ELEMENT_CHILD_- RELATIONSHIP	product_definition_relationship	41		
tdp_element_parent_to_tdp_- element_child_relationship to element_identification (child)	PATH			product_definition_relationship product_definition_relationship.related_product_definition->product_definition product_definition.formation->product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product_category product_category.name='document'

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
tdp_element_parent_to_tdp_- element_child_relationship to element_identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category product_category.name='document'
TDP_INDENTURED_ITEM	product_definition	41		
tdp_identured_item to item_- identification (top_indenture)	PATH			product_definition product_definition.formation-> product_definition_formation
TDP_INDENTURED_TDP_- ELEMENT	product_definition	41		
tdp_indentured_tdp_element to element_identification (top_indenture)	PATH			product_definition product_definition.formation-> product_definition_formation
TIME_INTERVAL_- EFFECTIVITY	time_interval_based_effectivity	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
time_interval_effectivity to date (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)
time_interval_effectivity to date (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)
time_interval_effectivity to event (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
time_interval_effectivity to event (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence
time_interval_effectivity to label (interval_name)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval time_interval.name
time_interval_effectivity to number_with_unit (duration)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.derivation-> measure_with_unit
VOLUME	measure_with_unit	41		

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
volume to number_with_units (a_value)	PATH			[measure_with_unit.unit_component-> unit unit=named_unit named_unit=> volume_unit] [named_unit.dimensions-> dimensional_exponents] {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-0} {dimensional_exponents.mass_exponent=0} {dimensional_exponents.length_exponent=3} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}
WEIGHT	representation_item	43		
weight to a_real (a_value)	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value measure_value=numeric_measure

Table 5 — Mapping table for data_definition_exchange (continued)

Application element	AIM element	Source	Rules	Reference path
weight to measure (unit_of_measure) #1: If weight is expressed in SI units #2: If weight is not expressed in SI units	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.unit_component-> unit unit=named_unit named_unit=> #1: (si_unit si_unit.name-> si_unit_name si_unit_name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]) #2: (conversion_based_unit conversion_based_unit.name [conversion_based_unit.conversion_factor-> measure_with_unit] [measure_with_unit.value_component-> measure_value] [measure_with_unit.unit_component-> unit unit=named_unit named_unit=> si_unit si_unit.name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]])

Table 5 - Mapping table for data_definition_exchange (concluded)

Application element	AIM element	Source	Rules	Reference path
weight to weight_derivation (derivation_method)	PATH			representation_item=> [qualified_representation_item qualified_representation_item.qualifiers[i]-> value_qualifier] [measure_representation_item<= measure_with_unit<- measure_qualification.qualified_measure measure_qualification {measure_qualification.name='weight derivation method'} measure_qualification.qualifier[i]-> value_qualifier] value_qualifier=type_qualifier type_qualifier.name {(type_qualifier.name='calculated') (type_qualifier.name='estimated') (type_qualifier.name='weighed')}

Table 6 — Mapping table for data_list

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_IDENTIFICATION_-OF_ITEM	product_definition_formation	41		
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ASSOCIATED_LIST #1: If the associated list is a parts_-list #2: If the associated list is a data_list #3: If the associated list is an other_list #4: If the associated list is an indentured data list #5: If the associated list is an index list	product_definition_formation	41		product_definition_formation product_defintion_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category #1:({product_category.name='parts list'}) #2:({product_category.name='data list'}) #3:({product_category.name='other list'}) #4:({product_category.name='indentured data list'}) #5:({product_category.name='index list'})
associated_list to drawing (related_to)	PATH			product_definition_formation <- product_defnition_formation_relationship.related_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='associated list'} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
associated_list to list_presentation (presentation)	PATH			product_definition_formation presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set
associated_list to product_data_set (related_to)	PATH			product_defination_formation<- product_defination.formation product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_defnition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name ='product data set'}

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
CONTRACT_SUBMISSION	applied_contract_assignment	232		{applied_contract_assignment<= contract_assignment role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='contract submission'}
contract_submission to contract (delivered_contract)	PATH			applied_contract_assignment<= contract_assignment contract_assignment.assigned_contract-> contract

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to date (date_of_submission) #1: If the date includes date and time #2: If the date only includes a date	PATH		5	applied_contract_assignment #1: (date_and_time_item=applied_contract_assignment date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role->date_time_role date_time_role.name='date and time of submission'} date_and_time_assignment.assigned_date_and_time->date_and_time) #2: (date_item=applied_contract_assignment date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment date_assignment.role->date_role date_role.name='date of submission'} date_assignment.assigned_date->date)

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to text (location)	PATH		6	applied_contract_assignment organization_item=applied_contract_assignment organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='location of contract submission'} organization_assignment.assigned_organization-> organization<- organizational_address.organizations[i] organizational_address<= address
DATA_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='data list'}}
data_list to data_list_body (body)	PATH			product_definition_formation<- product_defnition.formation product_definition

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list to data_list_header (list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='data list header'}
DATA_LIST_BODY	product_definition	41		{product_definition product_definition.description='data list body'}
data_list_body to data_list_tabulation (data_list_tabulations)	PATH			product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='data list tabulation'} product_definition_relationhsip.releted_product_definition-> product_definition
data_list_body to notation (data_list_notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition {property_definition.description='data list body'} property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_body to revision (revision_history)	PATH			product_definition <product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.\ relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action => {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
DATA_LIST_ENTRY #1: If the data_list entry is a AP 232 identified drawing, externally defined drawing, AP 232 identified sheet, externally defined sheet, other_list, parts_list, product_data_-set, data_list or reference_document. #2: If the data_list_entry is an AP 202 defined drawing, #3: If the data_list entry is an AP 202 defined sheet	#1: (product_definition) #2: ([draughting_drawing_revision] [product_definition]) #3: ([drawing_sheet_revision] [product_-definition])	41 505 41 505 41		{(product_definition product_definition.frame_of_reference->) (product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference->) product_definition_context product_definition_context<= application_context_element application_context_element..name='data list entry'}
data_list_entry to contract_-submission (available_from)	#1,#2,#3: (PATH)			#1,#2,#3: (product_definition contract_item=product_definition contract_item<- applied_contract_assignment.items[i] applied_contract_assignment)

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to header (list_entry) #3: (see note 2)	#1: (PATH) #2: (PATH)			#1: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition=property_definition property_definition #2: ([product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition=property_definition property_definition] [product_definition product_definition.formation product_definition_formation <product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='drawing'}> presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presentation_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set=> drawing_revision=> draughting_drawing_revision])

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to identifier (entry_item_change_level)	#1,#2,#3: (PATH)			#1,#2,#3: (product_definition action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change_identification for entry'} action_assignment.assigned_action-> action {action=> executed_action} action.name)
data_list_entry to notation (entry_notes)	#1,#2,#3: (PATH)			#1,#2,:#3: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition<- {property_definition.name='data list entry property'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item)

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to reference_- document (available_from)	#1,#2,#3: (PATH)			#1,#2, #3: (product_definition document_reference_item=product_definition document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='source identification'} document_reference.assigned_document-> document<- {document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name='document'})

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to reference_- document (list_entry) #2: (see note 2) #3: (see note 2)	#1: (PATH)			#1:(product_definition product_definition.formation->product_definition_formation)

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to sheet (list_entry) #2: (see note 2)	#1: (PATH) #3: (PATH)			#1: (product_definition product_definition.formation-> product_definition_formation <product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='sheet'}>) #3: (product_definition product_definition.formation product_definition_formation <product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='sheet'}> presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_representation presentation_representation=> presentation_area=> drawing_sheet_revision)

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_entry to source_location (available_from)	#1, #2, #3: (PATH)			#1, #2, #3:product_definition external_identification_item=product_definition external_identification_item<- applied_external_identificatoin_assignmant.items[i] applied_external_identification_assignment
data_list_entry to special_condition (special_conditions)	#1,#2,#3: (PATH)			#1,#2, #3: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition<- {property_definition.name='data list entry property'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item=> descriptive_representation_item)
DATA_LIST_HEADER	property_definition	41		{property_definition property_definition.description='data list header'}
data_list_header to header (common_header)	IDENTICAL MAPPING			
DATA_LIST_TABULATION #1: If the data_list_tabulation identifies the item that the data list is being tabulated for #2: If the data_list_tabulation does not identify what item the data list is being tabulated for	#1,#2: (product_definition)	41		

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
data_list_tabulation to data_list_entry (data_list_entries)	PATH			product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association {document_product_association=> document_product_equivalence) document_product_association.relating_document-> document<- {document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='data list entry'} applied_document_reference applied_document_reference.item[i]-> document_reference_item document_reference_item=product_definition product_defintion
data_list_tabulation to item_-identification (for_item) #2: (see note 2)	#1: (PATH)			#1: (product_definition product_definition.formation-> product_definition_formation)
DOCUMENT_LIST	product_definition_formation	41		

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 202 or AP 201 exchange with an AP 232 exchange	#1: (product_definition_formation) #2: ([draughting_drawing_revision] [product_definition_formation])	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'})

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to header (heading)	#1: (PATH) #2: (PATH)			#1, #2: (product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='drawing header'})
drawing to sheet (pages) (see note 3)				
DRAWING_SUFFIX_NUMBER_- COMBINATION	product	41		
drawing_suffix_number_- combination to element_- identification (drawing_number)		41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=\ 'drawing suffix number combination'} product_relationship.related_product->product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category.name='drawing'}

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_suffix_number_- combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name=\ 'drawing suffix number combination'}
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_- identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identification is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation) product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='alternate part identification') (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_relationship

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification-> certification
item_identification to change_-identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_-suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment=> organization_assignment {organization_assignment organization_assignment.role->organization_role organization_role.name='code administator'} organization_assignment.assigned_organization->organization
item_type to label (item_code)	PATH			product_category product_category.name

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.items[i] representation representation.name='notation'}

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role->classification_role classification_role.name='reference code'} classification_assignment.assigned_class->group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role->classification_role classification_role.name='type of notation'} classification_assignment.assigned_class->group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
OTHER_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='other list'}
other_list to other_list_body (list_body) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
other_list to other_list_header ((list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='other list header'}
OTHER_LIST_HEADER	property_definition	41		{property_definition property_definition.description='other list header'}
other_list_header to header (common_header)	IDENTICAL MAPPING			
PARTS_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='parts list'}
parts_list to parts_list_body (list_body) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
parts_list to parts_list_header (list_header)	PATH			product_definition_formation product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_defintion.definition property_definition {property_definition.description='parts list header'}
PARTS_LIST_HEADER	property_definition	41		{property_definition property_definition.description='parts list header'}
parts_list_header to header (common_header)	IDENTICAL MAPPING			
PRODUCT_DATA_SET	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'}
product_data_set to drawing (related) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to product_data_set (related) (see note 3)				
product_data_set to source_file (file_configuration) (see note 3)				
PRODUCT_DATA_SET_WITH_-FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set with formation'}
product_data_set_with_format to header (product_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='product data set with format header'}
product_data_set_with_format to product_presentation (presentation_of_product_data_set) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_WITH_-SHADING	product_definition_formation		24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set with shading'}
product_data_set_with_shading to header_configuration_with_-element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set with shading header'}
product_data_set_with_shading to shaded_shape_model (a_shaded_-model) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_- WITHOUT_FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set without formation'}
product_data_set_without_format to header_configuration_with_- element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set without format header'}
product_data_set_without_format to shape_model (a_model) (see note 3)				

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')}

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly with Revision when approval of person or organization is not needed. #2: date and time associated directly with Revision when approval of person or organization is not needed. #3 when approval of person or organization is needed and the date of their approval.	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to other_reference_- document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to person_and_organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {representation_item.name='size format') {representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
SINGLE_DOCUMENT_LIST	product_definition_formation	41		
SOURCE_LOCATION	applied_external_identification_- assignment	232		
source_location to text (path_information)	PATH			applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')}] [external_identification_assignment.source->external_source external_source.source_id]
source_location to text (storage_role_identification)	PATH			applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')} identification_assignment.assigned_id
SPECIAL_CONDITION	descriptive_representation_item	45		

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
special_condition to label (code)	PATH			descriptive_representation_item<= representation_item representation_item.name
special_condition to text (description)	PATH			descriptive_representation_item descriptive_representation_item.description
special_condition to text (type_of_coding_scheme)	PATH			descriptive_representation_item<= representation_item<- representation.items[i] representation {representation.name='special_condition'} representation.context_of_items[i]-> representation_context representation_context.context_type
SPECIFICATION_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='specification document'}

Table 6 — Mapping table for data_list (continued)

Application element	AIM element	Source	Rules	Reference path
STANDARD_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.producst[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='standard document'}

Table 6 - Mapping table for data_list (concluded)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a parts list #2: When tdp_element is a drawing defined within this document #3: When tdp_element is a product_-data_set #4: When tdp_element is a reference_document #5: When tdp_element is a data_list #6: When tdp_element is an other_-list #7: When tdp_element is a drawing or a product_data_set and is defined externally to this document #8: When tdp_element is a drawing that is defined by AP 202	#1, #2, #3, #4, #5, #6, #7: (product_-definition_formation) #8: (<draughting_drawing_revision> <product_definition_formation>)	41 505	24	{product_definiton_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {;roduct_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category #1: (product_category.name='parts list') #2: (product_category.name='drawing') #3: (product_category.name='product data set') #4: (product_category.name='reference document') #5: (product_category.name='data list') #6: (product_category.name='other list') #7: ((product_category.name='drawing') (product_category.name='product data set')) #8:(<product_category.name='drawing'> <draughting_drawing_revision>)}

Table 7 — Maping table for drawing

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_IDENTIFICATION_-OF_ITEM	product_definition_formation	41		
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product-> product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ANNOTATION	annotation_occurrence	46		
annotation to geometry (annotation_geometry)	PATH			annotation_occurrence<= styled_item {styled_item<= representation_item<- representation.items[i] representation} styled_item.item-> representation_item=> geometric_representation_item

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 202 or AP 201exchange with an AP 232 exchange	#1: product_definition_formation #2: [draughting_drawing_revision] [product_definition_formation]	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'})

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to header (heading) #2: (see note 2)	#1: (PATH)			#1: (product_definition_formation<- product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='drawing header'})

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to sheet (pages) #1: If drawing and sheet are identified within AP 232 #2: If drawing is identified within AP 232 and sheet is defined as AP 202/201 sheet within the same exchange #3: If drawing is defined as an AP 202/201 drawing and the sheet is identified within AP 232 Note: For drawing and sheet defined within AP 201 or AP 202, the mapping path can be found within AP 201 or AP 202, respectively	#1: (PATH) #2: (PATH) #3: (PATH)			#1: (product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'drawing membership'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: (product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				product_category.name='drawing'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'drawing membership'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'} presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_representation presentation_representation=> presentation_area=> drawing_sheet_revision) #3: (draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=>

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'drawing membership'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING_SUFFIX_NUMBER_-COMBINATION	product	41		
drawing_suffix_number_-combination to element_-identification (drawing_number)		41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=\ 'drawing suffix number combination'} product_relationship.related_product->product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category.name='drawing'}
drawing_suffix_number_-combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name=\ 'drawing suffix number combination'}
GEOMETRY	geometric_representation_item	42		

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification->certification

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to change_- identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment=> organization_assignment {organization_assignment organization_assignment.role->organization_role organization_role.name='code administator'} organization_assignment.assigned_organization->organization
item_type to label (item_code)	PATH			product_category product_category.name

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description
PRODUCT_PRESENTATION #1: For general capability #2: For use with AIC 517	#1: (presentatation_representation) #2: (mechanical_design_geometric_- presentation_representation)	46 517		
product_presentation to annotation (annotation_presentation) #2: (see note 2)	#1: (PATH)			#1: (presentation_representation<= representation representation.items[i]->) representation_item=> mapped_item mapped_item.mapping_source-> represenntaton_map mapped_representation-> representation representation.items[i]-> styled_item=> annotation_occurence)

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
product_presentation to shape_model (part_shape_views)	PATH			#1: (presentation_representation<= representation representation.items[i]->representation_item=> mapped_item mapped_item.mapping_source->representation_map representation_map.mapped_presentation->representation <representation=> shape_representation. shape_representation.context_of_items[i]->representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]->uncertainty_measure_with_unit> (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation) (shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation) (shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation))

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation=> shape_representation) (representation representation.items[i]-> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]-> advanced_face)) #2: (mechanical_design_geometric_presentation_\ representation<= presentation_representation<= representation representation.items[i]-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_presentation-> representation (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation) (shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation)

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation)) (representation=> shape_representation) (representation representation.items[i]->> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]->> advanced_face))
SHAPE_MODEL	#1: (edge_based_wireframe_shape_- representation) #2: (shell_based_wireframe_shape_- representation) #3: (geometrically_bounded_2d_- wireframe_representation) #4: (geometrically_bounded_surface_- shape_representation) #5: (non-manifold_surface_shape_- representation) #6: (manifold_surface_shape_- representation) #7: (geometrically_bounded_wireframe_- surface_shape_representation) #8: (advanced_face) #9: (faceted_brep_shape_representation) #10: (elementary_brep_shape_- representation) #11: (advanced_brep_shape_- representation) #12: (csg_shape_representation) #13: (shape_representation)	501 502 503 507 508 509 510 511 512 513 514 515 41		

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
shape_model to geometric_- validation_property (validation_properties)	PATH			representation<- {representation=> shape_representation} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> represented_definition represented_definition=property_definition property_definition {property_definition.name='geometric validation property'} <property_definition.definition-> characterzed_definition characterized_definition=shape_definition shape_definition #1: (shape_definition=product_definition_shape product_definition_shape) #2: (shape_definition=shape_aspect shape_aspect)> property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
shape_model to geometry (part_shape_representation)	PATH			#1: (edge_based_wireframe_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #2: (shell_based_wireframe_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #3: (geometrically_bounded_2d_wireframe_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #4: (geometrically_bounded_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #5: (non-manifold_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #6: (manifold_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #7: (geometrically_bounded_wireframe_surface_shape_representation<= shape_represenation<= representation representation.items[i]->

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item=> geometric_representation_item) #8: (advanced_face<- connected_face_set.cfs_faces[i]-> connected_face_set<= topological_representation_item<= representation_item=> geometric_representation_item) #9: (faceted_brep_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #10: (elementary_brep_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #11: (advanced_brep_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #12: (csg_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #13:(shape_representation<= representation representation.items[i]-> representation_item=> geometric_representation_item)

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 7 — Mapping table for drawing (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) Note: #1, #2: This part of ISO 10303 does not represent product presentation information for the sheet. Usage of ISO 10303:201 or ISO 10303:202 is required. #3: The mapping from drawing_-sheet_revision to the product information can be found in ISO 10303:201 or ISO 10303:202.				

Table 7 - Mapping table for drawing (concluded)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a product_-data_set #2: When tdp_element is a reference_document #3: When tdp_element is a drawing	product_definition_formation	41	24	#1:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set' }) #2:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='reference document'}) #3:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='drawing'})

Table 8 — Mapping table for indentured_data_list

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_IDENTIFICATION_-OF_ITEM	product_definition_formation	41		
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name=’id owner’)})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name=’creator’)}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name=’creator’)})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
ALTERNATE_ITEM #1: If alternate is a substitute with no concern for which is used in an assembly context #2: If alternate is an alternate item with concern for which is used in an assembly context #3: If alternate is an alternate item with a one for one substitution and allows for two way substitution in a particular assembly context	product	41		#1: ({product<- alternate_product_relationship.alternate}) #2: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute}) #3: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute})

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to an_integer (preference_order) #1: (see note 2) #3: (see note 2)	#2: (PATH)			#2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute assembly_component_usage_substitute_with_ranking assembly_component_usage_substitute_with_ranking.ranking)
alternate_item to item (interchange_item) #2: (see note 2) #3: (see note 2)	PATH			product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition
alternate_item to item_usage (interchange_item) #1: (see note 2) #3: (see note 2)	PATH			#2: product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- (product_definition_relationship.related_product_definition product_definition_relationship)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to label (type_of_alternate)	PATH			#1: (product<- alternate_product_relationship.alternate alternate_product_relationship alternate_product_relationship.name) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute assembly_component_usage_substitute.name) #3: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product_definition_substitute.name)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to text (usage_conditions) #4:Provides document or portion of document to describe occurrence relationship #5 Multiple usage_conditions can be captured in one text string and parsed, if needed, based on an exchange partner business agreement.	#1: (PATH) #2: (PATH) #3: (PATH)			#1: ({<product<- alternate_product_relationship.basis> <product<- alternate_product_relationship.description> <alternate_product_relationship document_reference_item=alternate_product_relationship document_reference_item<- (applied_document_reference.items[i] applied_document_reference applied_document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document>}) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute #4: <document_reference_item=assembly_component_usage_substitution document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				document> assembly_component_usage_substitute.definition) #3: (product<- product_definition_formation.of_product product_definition_formation<- jproduct_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product-definition-substitute.description)
ASSOCIATED_LIST #1: If the associated list is a parts_-list #2: If the associated list is a data_list #3: If the associated list is an other_list #4: If the associated list is an indentured data list #5: If the associated list is an index list	product_definition_formation	41		product_definition_formation product_defintion_formation.of_product-> product<- product_related_prodcut_category.products[i] product_related_product_category<= product_category #1:({product_category.name='parts list'}) #2:({product_category.name='data list'}) #3:({product_category.name='other list'}) #4:({product_category.name='indentured data list'}) #5:({product_category.name='index list'})
associated_list to drawing (related_to)	PATH			#1: (product_definition_formation <- product_defnition_formation_relationship.\ related_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='associated list'} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
associated_list to list_presentation (presentation)	PATH			product_definition_formation presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set
associated_list to product_data_set (related_to)	PATH			product_defination_formation<- product_defination.formation product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_defnition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name ='product data set'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
CONTENT_PROPERTY	representation	43		{representation<- {[representation.name='document content'] [representation.context_of_items-> representation_context representation_context.context_type='document parameters']} property_definition_representation.used_representation property_definition_representation property_definition_representation.defintion-> represented_definition represented_definition=property_definition property_definition {property_definition.name='document property'}}
content_property to a_real (real_world_scale) #1: real number value #2: text number	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='real world scale'} #1: (measure_representation_item<= measure_with_unit [measure_with_unit.value_component-> measure_value measure_value=ratio_measure] [measure_withunit.unit_component-> unit unit=named_unit named_unit-> ratio_unit]) #2: (descriptive_representation_item descriptive_representation_item.description)
content_property to text (detail_level)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='detail level'} descriptive_representation_item descriptive_representation_item.descrption

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
content_property to text (geometry_type)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='geometry type'} descriptive_representation_item descriptive_representation_item.description
content_property to text (languages) #1: through descrptive text #2: through classification structure	PATH			#1: (representation representation.items[i]-> representation_item=> {representation_item.name='language'} descriptive_representation_item.description) #2: (representation language_item=representation language_item<- language_assignment.items[i] language_assignment<= classification_assignment {classification_assignment.role-> classification_role classification_role.name='language'} classification_assignment.assigned_class-> group {group=> language} group.name)
CONTRACT_SUBMISSION	applied_contract_assignment	232		{applied_contract_assignment<= {contract_assignment role_select=contract_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='contract submission'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to contract (delivered_contract)	PATH			applied_contract_assignment<= contract_assignment contract_assignment.assigned_contract-> contract
contract_submission to date (date_of_submission) #1: If the date includes date and time #2: If the date only includes a date	PATH		5	applied_contract_assignment #1: (date_and_time_item=applied_contract_assignment date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role-> date_time_role date_time_role.name='date and time of submission'} date_and_time_assignment.assigned_date_and_time-> date_and_time) #2: (date_item=applied_contract_assignment date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment date_assignment.role-> date_role date_role.name='date of submission'} date_assignment.assigned_date-> date)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
contract_submission to text (location)	PATH		6	applied_contract_assignment organization_item=applied_contract_assignment organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='location of contract submission'} organization_assignment.assigned_organization-> organization<- organizational_address.organizations[i] organizational_address<= address
DATE_EFFECTIVITY	dated_effectivity	41		
date_effectivity to date (end_date)	PATH			dated_effectivity dated_effectivity.effectivity_end_date-> date_and_time
date_effectivity to date (start_date)	PATH			dated_effectivity dated_effectivity.effectivity_start_date-> date_and_time
DOCUMENT_LIST	document	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or 202 exchange with an AP 232 exchange	#1: (product_definition_formation) #2: ([draughting_drawing_revision] [product_definition_formation])	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'})

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to header (heading)	#1: (PATH) #2: (PATH)			#1, #2: (product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='drawing header'})
drawing to sheet (pages) (see note 3)				
DRAWING_SUFFIX_NUMBER_- COMBINATION	product	41		
drawing_suffix_number_- combination to element_- identification (drawing_number)	PATH	41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=\ 'drawing suffix number combination'} product_relationship.related_product->product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category.name='drawing'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_suffix_number_- combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name=\ 'drawing suffix number combination'}
EFFECTIVITY #1: When the effectivity is for a part and the next higher part is known #2: When the effectivity is for a part and the next higher part is not known	#1: (<configuration_effectivity> <(serial_numbered_effectivity) (dated_effectivity) (lot_effectivity)>) #2: ([applied_effectivity_assignment] [applied_effectivity_context_- assignment}	41 41 41 41 232 232		#1: ({configuration_effectivity<= product_definition_effectivity product_definition_effectivity.usage->product_definition_relationship product_definition_relationship.description='item usage'}) #2: ({applied_effectivity_assignment<= effectivity_assignment<- effectivity_context_assignment.assigned_effectivity_assignment effectivity_context_assignment [effectivity_context_assignment.role->effectivity_context_role effectivity_context_role.name='context document of effectivity'] [effectivity_context_assignment=> applied_effectivity_context_assignment applied_effectivity_context_assignment.items[i]->effectivity_context_item effectivity_context_item=product_definition_formation product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product.category product_category.name='document']})

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to approval (effectivity_approvals)	PATH			#1: (configuration_effectivity approval_item=configuration_effectivity approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) #2: (applied_effectivity_assignment approval_item=applied_effectivity_assignment approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role>
effectivity to label (name)	PATH			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.named_item name_attribute name_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.described_item name_attribute name_attribute.attribute_value)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to label (work_activity)	PATH			configuration_effectivity action_item=configuration_effectivity action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.name
effectivity to product_configuration (product)	PATH			#1: (configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity=> product_definition_effectivity=> configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to text (description)	PATH			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attribute.described_item description_attribute description_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attribute.described_item description_attribute description_attribute.attribute_value)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
EVENT	event_occurrence	41		
event to date (actual_date) #1: Just date #2: date and time	PATH			event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name='actual date'} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name='actual date'} date_assignment.assigned_date_and_time-> date_and_time)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
event to date (planned_date)	PATH			event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name='planned date'} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name='planned date'} date_assignment.assigned_date_and_time-> date_and_time)
event to identifier (id)	event_occurrence.id	41		
event to label (name)	event_occurrence.name	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
event to person_and_organization (responsible_person_organization)	PATH			event_occurrence (person_and_organization_item=event_occurrence person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role person_and_organization_role.name='identifier'} person_and_organization_assignment.assigned_person_and_organization-> person_and_organization) (organization_item=event_occurrence organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role organization_role.name='system destination'} organization_assignment.assigned_organization-> organization)
event to project (assignment)	PATH			event_occurrenc<- event_occurrence_assignment.assigned_event_occurrence event_occurrence_assignment<= applied_event_occurrence_assignment applied_event_occurrence_assignment.items[i]-> event_occurrence_item event_occurrence_item=organizational_project organizational_project
event to text (description)	event_occurrence.description	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
EXTERNAL_LIBRARY_- REFERENCE #1: If external_library_reference is an element in a class source select, that is Reference by a general_-classification as the classification_-source. #2: If external_library_reference is an element in a property source select, that is reference by a property as the property_source.	#1: (externally_defined_class) #2: (externally_defined_general_-property)	232 232		#1: (externally_defined_class<= [class<= group][externally_defined_item]) #2:(externally_defined_general_property<= [general_property] [externally_defined_item])
external_library_reference to identifier (external_id)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.item_id
external_library_reference to identifier (library_type)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source-> external_sourrce external_sourrce.id {external_source.source_id-> source_item source_item=identifier identifier
external_library_reference to text (description)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.cource-> exterrnal_source external_sourcee.description

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
INDENTURED_DATA_LIST	product_definition_formation	41	19 24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='indentured data list'}
indentured_data_list to indentured_-data_list_body (body)	PATH			product_definition_formation<- product_definition.formation (product_definition)(product_definition_with_associated_documents) {product_definition.description=’indentured data list content’}
indentured_data_list to indentured_-data_list_header (header)	PATH			product_definition_formation<- product_definition.formation (product_definition)(product_definition_with_associated_documents) {product_definition.description=’indentured data list content’} characterized_product_definition=(product_defnition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=’indentured data list header’}
INDENTURED_DATA_LIST_-BODY	product_definition	41		{product_definition product_definition.description=‘indentured data list content'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_body to element identification (standardization_documents_list)	PATH			product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association {document_product_association=> document_product_equivalence} document_product_association.relating_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='standards document'} applied_document_reference applied_document_reference.items[i]-> document_reference_item document_reference_item=product_definition_formation product_definition_formation {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category product_category.name='document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_body to indentured_data_list_tabulation (indentured_tabulation)	PATH		21	product_definition product_or_formation_or_definition=product_definition product_or_formation_or_definition<- document_product_association.related_product document_product_association {document_product_association=> document_product_equivalence} {document_product_association.name='equivalence'} document_product_association.relating_document-> document<- {document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document definition'} document_reference.assigned_document document_reference=> {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by document') (object_role.name='indentured by item') (object_role.name='indentured by item and document')} applied_document_reference

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_body to notation (notes_list)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='indentured data list body property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_body to revision (revision_history)	PATH			product_definition <product_definition_formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.\ relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action => {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
INDENTURED_DATA_LIST_- ENTRY #1: When entry is a document #2: When entry is an item	#1,#2: (product_definition_with_- associated_documents) (product_definition)	41		{(product_definition (product_definition.frame_of_reference-> product_definition_context) (product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference-> product_definition_context) product_definition_context=> application_context_element application_context_element.name='indentured data list entry'}
indentured_data_list_entry to configuration (entry_configuration)	PATH			#1,#2:(product_definition_with_associated_documents)\ (product_definition) characterized_product_definition=\ (product_definition_with_associated_documents)(product_definition) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {(property_definition.name) (property_definition.name='indentured data list entry property')}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to content_property (entry_content_property)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name=document content'} {representation.context_of_items-> representation_context representation_context.context_type='document parameters'}
indentured_data_list_entry to contract_submission (available_from)	PATH			(product_definition_with_associated_documents)\ (product_definition) contract_item=product_definition contract_item<- applied_contract_assignment.items[i] applied_contract_assignment)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to effectivity (effective_on) #3: Provide effectivity based on usage #4: Provide effectivity for view or instance. #5: Provide effecitivity for version #6: Provide effecitivity based on item instance or relationship	PATH			#3: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship<- product_definition_effectivity.usage product_definition_effectivity=> configuration_effectivity) (#4: (prodcut_definition effectivity_item=product_definition) #5: (product_definition product_definition.formation-> product_definition_formation effectivity_item=product_definition_formation) #6: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship effectivity_item=product_definition_formation) effectivity_item<- applied_effectivity_assignment.item[i] applied_effectivity_assignment=> effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to element_identification (superseded _entry) #1: Identifies superseded element through a definition view relationship #2: Identifies superseded element through a version relationship #3: Identifies superseded element through change request information. The latest item can be identified through its version, view, or structured relationship. The element being superseded can also be identified through its version, view, or structured relationship. #4: Provides approval for superseded request #5: Provides proposed solution to request. #6: Provides status for the request. #7: Provides date and time for the request. #8: Provides date and time for when the superseded activity was applied. #9: Provides document or portion of document to describe the superseded change. #10: Provides who requested the change. #11: Provides what project this work acitivity is associated with #12: Identifies resourcesto perform the activity. #13: Identifies properties that are being requested.	PATH		7	#1: (product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='superseded element'} product_definition_relationship.related_product_definition -> product_definition product_definition.formation-> product_definition_formation) #2: (product_definition-> product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship #9: <document_reference_item=product_definition_formation_relationship document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> {product_definition_formation_relationship.name=\ 'superseded version'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #3: ((product_definition action_item=product_definition) (product_definition product_definition.formation-> product_definition_formation action_item=product_definition_formation) (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship action_item=product_definition_relationship) action_item<-

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_asignment.items[i] applied_action_assignment<= #8: <applied_action_assignment (date_item=applied_action_assignment date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (date_and_time_item=applied_action_assignment date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> action_assignment action_assignment.assigned_action-> action=> #11:<actioin organizational_project_item=action organizational_project_item<- applied_organizational_project_assignment.items[i] applied_organizational_project_assignment<= organizational_project_assignment organizational_project_assignment.assigned_organizational_project-> organizational_project> #12: <action source_item=action source_item<- action_resource.usage[i] action_resource<- action_resource_relationship.related_resource action_resource_relationship> executed_action=> directed_action directed_action.directive-> action_directive action_directive.requests[i]->

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				versioned_action_request<- #4:<versioned_action_request approval_item=versioned_action_request approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval> #5: <versioned_action_request<- action_request_solution.request action_request_solution> #6: <versioned_action_request<- action_request_status.assigned_request action_request_status> #7: <(versioned_action_request date_item=versioned_action_requrest date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (versioned_action_request date_and_time_item=versioned_action_requrest date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> #9: <versioned_action_request document_reference_item=versioned_action_request document_reference_item<- applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document-> document> #10:versioned_action_request

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(person_and_organization_item=versioned_action_request person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='requestor')}) (organization_item=versioned_action_request organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='requestor')})> #13 :<versioned_action_request<- action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment applied_aciton_request_assignment.items[i]-> action_request_item action_request_item=property_definition property_definition> action_request_assignement.assigned_action_request action_request_assignment=> applied_action_request_assignment applied_aciton_request_assignment.items[i]-> action_request_item (action_request_item=product_definition_formation product_definition_formation) (action_request_item=product_definition product_definition product_definition.formation-> product_definition_formation) (action_request_item=product_definition_relationship product_definition_relationship product_definition_relationship.related_product_definition->

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition product_definition.formation-> product_definition_formation))
indentured_data_list_entry to identifier (entry_item_change_level)	PATH			product_defnition action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification for entry'} action_assignment.assigned_action-> action<- {action=> executed_action} action_relationship.relationship_action action_relationship action_relationship.related_action-> action {action=> executed_action} action.name

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to item_-identification (superseded _entry) #1: Identifies superseded item through a definition view relationship #2: Identifies superseded item through a version relationship #3: Identifies superseded item through change request information. The latest item can be identified through its version, view, or structured relationship. The item being superseded can also be identified through its version, view, or structured relationship. #4: Provides approval for superseded request #5: Provides proposed solution to request. #6: Provides status for the request. #7: Provides date and time for the request. #8: Provides date and time for when the superseded activity was applied. #9: Provides document or portion of document to describe the superseded change. #10: Provides who requested the change. #11: Provides what project this work acitivity is associated with #12: Identifies resourcesto perform the activity. #13: Identifies properties that are being requested.	PATH			#1: (product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='superseded element'} product_definition_relationship.related_product_definition -> product_definition product_definition.formation-> product_definition_formation) #2: (product_definition-> product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship #9: <document_reference_item=product_definition_formation_relationship document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> {product_definition_formation_relationship.name=\ 'superseded version'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #3: ((product_definition action_item=product_definition) (product_definition product_definition.formation-> product_definition_formation action_item=product_definition_formation) (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship action_item=product_definition_relationship) ation_item<-

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_asignment.items[i] applied_action_assignment<= #8: <applied_action_assignment (date_item=applied_action_assignment date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (date_and_time_item=applied_action_assignment date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> action_assignment action_assignment.assigned_action-> action=> #11:<actioin organizational_project_item=action organizational_project_item<- applied_organizational_project_assignment.items[i] applied_organizational_project_assignment<= organizational_project_assignment organizational_project_assignment.assigned_organizational_project-> organizational_project> #12: <action source_item=action source_item<- action_resource.usage[i] action_resource<- action_resource_relationship.related_resource action_resource_relationship> executed_action=> directed_action directed_action.directive-> action_directive action_directive.requests[i]->

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				versioned_action_request<- #4:<versioned_action_request approval_item=versioned_action_request approval_item<- applied_approval_assignment.items applied_approval_assignment<= approval_assignment approval_assignment.assigned_approval-> approval> #5: <versioned_action_request<- action_request_solution.request action_request_solution> #6: <versioned_action_request<- action_request_status.assigned_request action_request_status> #7: <(versioned_action_request date_item=versioned_action_requrest date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment date_assignment.assigned_date-> date) (versioned_action_request date_and_time_item=versioned_action_requrest date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment date_assignment.assigned_date_and_time-> date_and_time)> #9: <versioned_action_request document_reference_item=versioned_action_request (action_request_item=product_definition_formation product_definition_formation) (action_request_item=product_definition product_definition product_definition.formation-> product_definition_formation) (action_request_item=product_definition_relationship

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation))
indentured_data_list_entry to item_- parent_to_item_child_relationship (entry)	PATH			product_definition<- product_definition_relationship.related_product_definition (product_definition_relationship)(alternate_product_relationship)\ (assembly_component_usage)\ (assembly_component_usage_substitute)\ (next_assembly_usage_occurrence)\ (quantified_assembly_component_usage)\ (assembly_component_usage_with_ranking)\ (make_from_usage_option_with_reference_designator)\ (make_from_usage_option)
indentured_data_list_entry to item_- parent_to_tdp_element_child_- relationship (entry)	PATH			((product_definition)(product_definition_with_associated_documents) product_definition_relationship.related_product_definition product_definition_relationship) ((product_definition)(product_definition_with_associated_documents) document_reference_item=(product_definition)/ (product_definition_with_associated_documents) document_reference_item<- applied_document_reference.items[i] applied_document_reference)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to label (indenture_level)	PATH		20	#1,#2: (product_definition characterized_product_definition=product_definition characterized_defnition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='indentured data list entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='indentured level tag'} representation.items[i]->representation_item {representation_item=>descriptive_representation_item} representation_item.name)
indentured_data_list_entry to notation (entry_notes)	#1: (PATH) #2: (PATH)			#1,#2: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='indentured data list entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='notation'} representation.items[i]->representation_item=>descriptive_representation_item)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to reference_document (available_from)	PATH			(product_definition)\ (product_definition_with_associated_documents) document_reference_item=(product_definition)\ (product_definition_with_associated_documents) document_reference_item<- applied_document_referenece.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='source identification'} document_reference.assigned_document-> document<- {docoument.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category product_category.name='document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to retrofit_usage (retrofit)	PATH			product_definition<- product_definition_relationship.related_product_definition product_definition_relationship action_item=product_definition_relationship action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='retrofit usage'}
indentured_data_list_entry to source_location (available_from)	PATH			product_definition external_identification_item=product_definition external_identification_item<- applied_external_identificatoin_assignmant.items[i] applied_external_identification_assignment

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to special_condition (special_conditions)	#1: (PATH) #2: (PATH)			#1,#2: (product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='indentured data list entry property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item=> descriptive_representation_item)
indentured_data_list_entry to tdp_-element_parent_to_item_child_-relationship (entry)	PATH			(product_definition)(product_definition_with_associated_documents)<- product_definition_relationship.related_product_definition product_definition_relationship
indentured_data_list_entry to tdp_-element_parent_to_tdp_element_-child_relationship (entry)	PATH			(product_definition)(product_definition_with_associated_documents)<- product_definition_relationship.related_product_definition product_definition_relationship
indentured_data_list_entry to tdp_-indentured_item (entry) #1: (see note 2)	#2: (IDENTICAL MAPPING)			

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_entry to tdp_-indentured_tdp_element (entry) #2: (see note 2)	#1: (IDENTICAL MAPPING)			
INDENTURED_DATA_LIST_-HEADER	property_definition			{property_definition property_definition.name='indentured data list header'}
indentured_data_list_header to header (common_header)	IDENTICAL_MAPPING			
indentured_data_list_header to item_identification (procurement_references)	PATH			property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.relating_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name=\ 'procurement reference'} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
INDENTURED_DATA_LIST_- TABULATION	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name='indentured by document') object_role.name='indentured by item') object_role.name='indentured by item and document')}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_tabulation to indentured_data_list_entry (tabulation_of_entries)	PATH			applied_document_reference applied_document_reference.items[i]->document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.related_product_definition-> product_definition> <document_reference_item=applied_document_reference applied_document_reference<= document_reference {applied_document_reference<= document_reference document_reference.source='child'} document_reference.assigned_document-> document<- document_product_association.relating_document document_product_association=> document_product_equivalence<= document_product_association document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_defintion_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition->

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference {applied_document_reference<= document_reference document_reference.source='parent'} applied_document_reference.item{i}-> document_reference_item (document_reference_item=product_definition product_definition) (document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition)>
indentured_data_list_tabulation to indentured_list_by_document (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_data_list_tabulation to indentured_list_by_part (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item'}
indentured_data_list_tabulation to indentured_list_by_part_with_-document_references_to_parts (method_of_entry_tabulation)	IDENTICAL MAPPING			{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item and document'}
INDENTURED_LIST_BY_-DOCUMENT	applied_document_reference	232		{applied_document_reference<= document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_document to tdp_indentured_tdp_element (top_indentured_tdp_elements)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition>
INDENTURED_LIST_BY_PART	applied_document_reference	232		{applied_document_reference<- document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part to tdp_- indentured_item (top_indentured_items)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition)>
INDENTURED_LIST_BY_PART_- WITH_DOCUMENT_- REFERENCES_TO_PARTS	applied_document_reference	232		{applied_document_reference<- document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='indentured by item and document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part_with_- document_references_to_parts to tdp_indentured_item (top_indentured)	PATH			applied_document_reference applied_document_reference.items[i]->document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation->product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition->product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]->document_reference_item (document_reference_item=product_definition product_definition) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition)>

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
indentured_list_by_part_with_-document_references_to_parts to tdp_indentured_tdp_element (top_indentured)	PATH			applied_document_reference applied_document_reference.items[i]-> document_reference_item <document_reference_item=product_definition_formation_relationship product_definition_formation_relationship product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation<- product_definition.formation product_definition> < document_reference_item=product_definition_relationship product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition> <document_reference_item=product_definition_formation product_definition_formation<- product_definition.formation product_definition> <document_reference_item=product_definition> <document_reference_item=applied_document_reference applied_document_reference applied_document_reference.items[i]-> document_reference_item (document_reference_item=product_definition product_definition) (document_reference_item=product_defintion_formation product_definition_formation<- product_definition.formation product_definition)>
INDEPENDENT_PROPERTY	general_property	41		
independent_property to external_-library_reference (property_source)	IDENTICAL MAPPING			{general_property=> externally_defined_general_property}
independent_property to identifier (id)	general_property.id	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property to measure (allowed_unit) #1: If the unit has a commonly used descriptor. #2: If the unit descriptor is constructed of more than one unit name.	PATH			general_property represented_definition=general_property represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='allowed units'} representation.context_of_item->representation_context=> global_unit_assigned_context.units[i]->unit #1: (unit=named_unit named_unit) #2: (unit=derived_unit derived_unit)
independent_property_to text (description)	general_property.description	41		
INDEPENDENT_PROPERTY_-RELATIONSHIP	general_property_relationship	41		
independent_property_relationship to text (description)	general_property_relationship.description	41		
independent_property_relationship to independent_property (related_property)	PATH			general_property_relationship general_property_relationship.related_property->general_property
independent_property_relationship to independent_property (relating_property)	PATH			general_property_relationship general_property_relationship.relating_property->general_property

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property_relationship to label (relation_type)	general_property_relationship.name	41		
INDEPENDENT_PROPERTY_-USAGE	general_property_association	41		
independent_property_usage to independent_property (assigned_property)	PATH			general_property_association general_property_association.base_definition-> general_property
independent_property_usage to property_assignment (assigned_to)	PATH			general_property_association general_property_association.derived_definition-> derived_property_select derived_property_select=property_definition property_definition
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name=('alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation-> product_definition_formation) #2:(product_definition_formation-> product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product-> product<- product_definition_formation)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification-> certification
item_identification to change_-identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_-information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
ITEM_PARENT_TO_ITEM_-CHILD_RELATIONSHIP	product_definition_relationship	41		
item_parent_to_item_child_-relationship to item_identification (child)	PATH			product_definition_relationship product_definition_relationship.related_product_definition->(product_definition)(product_definition_with_associated_documents) product_definition.formation->product_definition_formation
item_parent_to_item_child_-relationship to item_identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition->(product_definition)(product_definition_with_associated_documents) product_definition.formation->product_definition_formation
ITEM_PARENT_TO_TDP_-ELEMENT_CHILD_-RELATIONSHIP	(product_definition_relationship)(applied_document_reference)	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_parent_to_tdp_element_child_- relationship to element_identification (child)	PATH			(product_definition_relationship product_definition_relationship.related_product_definition-> (product_definition)(product_definition_with_associated_documents) product_definition.formation-> product_definition_formation) (applied_document_reference<= document_reference document_reference.assigned_document-> document<- {document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category product_category.name='document'})

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_parent_to_tdp_element_child_-relationship to item_identification (parent)	PATH			(product_definition_relationship product_definition_relationship.relating_product_definition-> (product_definition)(product_definition_with_associated_documents) product_definition.formation-> product_definition_formation) (applied_document_reference applied_document_reference.items[i]-> document_reference_item document_reference_item=product_definition_formation product_definition_formation)
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment=> organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='code administator'} organization_assignment.assigned_organization-> organization
item_type to label (item_code)	PATH			product_category product_category.name
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
LOT_EFFECTIVITY	lot_effectivity	41		
lot_effectivity to a_number (lot_size)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.value_component-> measure_value
lot_effectivity to identifier (lot_identification)	PATH			lot_effectivity lot_effectivity.effectivity_lot_id
lot_effectivity to measure (lot_unit_of_measure)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.unit_component-> unit
MEASURE	named_unit	41		
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.items[i] representation representation.name='notation'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role_> classification_role classification_role.name='reference code'} classification_assignment.assigned_class-> group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role-> classification_role classification_role.name='type of notation'} classification_assignment.assigned_class-> group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
NUMBER_WITH_UNITS	measure_with_unit	41		
number_with_units to a_real (value_of)	PATH			measure_with_unit measure_with_unit.value_component measure_value
number_with_units to measure (units_of)	PATH			measure_with_unit measure_with_unit.unit_component unit
PRODUCT_DATA_SET	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to drawing (related) #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or AP 202 exchange within the same file	PATH			#1:(product_definition_formation<- {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related drawing'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'}) #2: (product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related drawing'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing'} presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set=> drawing_revision=> draughting_drawing_revision)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to product_data_set (related)	PATH			product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related product data set'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to source_file (file_configuration)	PATH			product_definition_formation<- product_definition.formation (product_definition=> product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i]-> document=> document_file) (product_defintion document_reference_item=product_defintion document_reference_item<- applied_document_reference.item[i] applied_document_reference<= document_reference=> document_file)
PRODUCT_DATA_SET_WITH_-FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set with format'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_with_format to header (product_header)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name=\ 'product data set with format header'}
product_data_set_with_format to product_presentation (presentation_of_product_data_set) (see note 3)				
PRODUCT_DATA_SET_WITH_-SHADING	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- product_category.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set with shading'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_with_shading to header_configuration_with_-element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set with shading header'}
product_data_set_with_shading to shaded_shape_model (a_shaded_model) (see note 3)				
PRODUCT_DATA_SET_-WITHOUT_FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set without format'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_without_format to header_configuration_with_-element_identification (data_configuration)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='product data set without format header'}
product_data_set_without_format to shape_model (a_model) (see note 3)				

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
PROJECT	organizational_project	41		
project to date (end_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role->date_role (date_role.name) (date_role.name='actual end') (date_role.name='planned end')} date_assignment.assigned_date->date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role->date_time_role (date_time_role.name) (date_time_role.name='actual end') (date_time_role.name='planned end')} date_and_time_assignment.assigned_date->date_and_time)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
project to date (start_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role->date_role (date_role.name) (date_role.name=’actual start’) (date_role.name='planned start')} date_assignment.assigned_date->date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role->date_time_role (date_time_role.name) (date_time_role.name=’actual start’) (date_time_role.name='planned start')} date_and_time_assignment.assigned_date->date_and_time)
project to identifier (id)	organizational_project.id	41		organizational_project id_attribute_select=organizational_project id_attribute_select<- id_attribute.identified_item id_attribute id_attribute.attribute_value
project to label (name)	organizational_project.name	41		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
project to perrson_and_organization (participants)	PATH			organizational_project (person_and_organization_item=organizational_project person_and_organization_item<- applied_person_and_organization_assignment.item[i] applied_person_and_organization_assignment=> person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=person_and_organization person_organization_select) (organization_item=organizational_project organization_item<- applied_organization_assignment.item[i] applied_organization_assignment=> organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
project to person_and_organization (project_owner)	PATH			organizational_project organizational_project.responsible_organization[i]-> organization person_organization_select=organization person_organization_select
project to text (description)	organizational_project.description	41		
PROPERTY_ASSIGNMENT	property_definition	41		
property_assignment to identifier (described_element)	PATH			property_definition property_definition.definition-> characterized_definition

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
property_assignment to label (name)	property_definition.name	41		
property_assignment to text (description)	property_definition.description	41		
QUANTITY	measure_with_unit	41		
quantity to a_number (value_of)	PATH			measure_with_unit measure_with_unit.value_component-> measure_value measure_value=count_measure count_measure
quantity to text (quantity_accuracy)	PATH			measure_with_unit<- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifier-> type_qualifier type_qualifier.name {(type_qualifier.name) (type_qualifier.name='exact') (type_qualifier.name='as required') (type_qualifier.name='approximate')}
quantity to text (value_of)	PATH			measure_with_unit measure_with_unit.value_compenent-> measure_value measure_value=descriptive_measure descriptive_measure
quantity to volume (value_of)	IDENTICAL MAPPING			

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
quantity to weight (value_of)	PATH			measure_with_unit=> measure_representation_item<= representation_item
RELATIVE_EVENT	relative_event_occurrence	41		
relative_event to event (related_event)	PATH			relative_event_occurrence relative_event_occurrence.base_event-> event_occurrence
relative_event to number_with_units (offset)	PATH			relative_event_occurrence relative_event_occurrence.offset-> time_measure_with_unit=> measure_with_units
RETROFIT_USAGE	applied_action_assignment	232		{applied_action_assignment<= action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='retrofit usage'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
retrofit_usage to notation (retrofit_description)	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action characterized_action_definition=action characterized_action_definition<- action_property.definition action_property<- action_property_representation.property action_property_representation {action_property_representation.name='retrofit notation'} action_property_representation.representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
retrofit_usage to other_reference_-document (defining_document) #1: reference full document #2: reference portion of document	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.chosen_method-> (action_method=> action_method_with_associated_documents action_method_with_associated_documents.specification->) (action_method document_reference_item=action_method document_reference_item<- #1:(applied_document_reference.item[i] applied_document_reference<= document_reference document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->)) document
retrofit_usage to retrofit_state (disposition_for)	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.chosen_method-> action_method action_method.name {(action_method.name) (action_method.name='add') (action_method.name='delete') (action_method.name='modify')}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification'))}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly to Revision when approval of person or organization is not needed. #2: date and time associated directly to Revision when approval of person or organization is not needed. #3 when approval of person or organization is needed and the date of their approval	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to other_reference_- document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to person_and_organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
SEQUENCE_EFFECTIVITY	serial_numbered_effectivity	41		
sequence_effectivity to identifier (from_effectivity_id)	PATH			serial_numbered_effectivity serial_numbered_effectivity.effectivity_start_id
sequence_effectivity to identifier (thru_effectivity_id)	PATH			serial_numbered_effectiivity serial_numbered_effectivity..effectivity_end_id
sequence_effectivity to measure (quantity_unit_of_measure)	PATH		10	serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit measure_with_unit.unit_component-> unit
sequence_effectivity to quantity (total_component_quantity_in_-product_configuration)	PATH			serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
SOURCE_INFORMATION_TYPE	representation	43	22	{representation representation.name='source information type'}
source_information_type to identifier (source_code)	PATH			representation representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} representation_item.name
source_information_type to text (source_description)	PATH			representation representation.items[i]-> representation_item=> descriptive_representation_item descriptive_representation_item.description
source_information_type to text (type_of_coding_scheme)	PATH			representation representation.context_of_items[i]-> representation_context representation_context.context_type
SOURCE_LOCATION	applied_external_identification_-assignment	232		

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
source_location to text (path_information)	PATH			applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role-> (identification_role) (identification_role.description='access context')}] [external_identification_assignment.source-> external_source external_source.source_id]
source_location to text (storage_-node_identification)	PATH			applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role-> (identification_role) (identification_role.description='access context')} identification_assignment.assigned_id
SPECIAL_CONDITION	descriptive_representation_item	45		
special_condition to label (code)	PATH			descriptive_representation_item<= representation_item representation_item.name
special_condition to text (description)	PATH			descriptive_representation_item descriptive_representation_item.description
special_condition to text (type_of_coding_scheme)	PATH			descriptive_representation_item<= representation_item<- representation.items[i] representation {representation.name='special condition'} representation.context_of_items[i]-> representation_context representation_context.context_type

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
SPECIFICATION_DOCUMENT	product_definition_formation		24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] prodcut_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference_document'} product_category_relationship.category product_category_relationship product_category-relationsip.sub_category-> product_category product_category.name='specification document'}
STANDARD_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] prodcut_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference_document'} product_category_relationship.category product_category_relationship product_category-relationsip.sub_category-> product_category product_category.name='standard document'}

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a reference_document #2: When tdp_element is an indentured_data_list #3: When tdp_element is a drawing or product_data_set defined externally to thisdocument #4: When tdp_element is a drawing that is defined by AP 202	#1,#2,#3: (product_definition_formation) #4: (<draughting_drawing_revision> <product_definition_formation>)	41 505 41	24	#1:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='reference document' }) #2:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='indentured data list' }) #3:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category (product_category.name='drawing') (product_category.name='product data set') })

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
				#4:(<draughting_drawing_revision> {product_definition_formation product_definition_formation.of_product -> product<- product_related_product_category.products[i] product_related_product_category<= product_category product_category.name='document' product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='drawing' })
TDP_ELEMENT_PARENT_TO_- ITEM_CHILD_RELATIONSHIP	product_definition_relationship	41		
tdp_element_parent_to_item_child_- relationship to element_identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category product_category.name='document'
tdp_element_parent_to_item_child_- relationship to item_identification (child)	PATH			product_definition_relationship product_definition_relationship.related_product_definition -> product_definition product_definition.formation-> product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT_PARENT_TO_- TDP_ELEMENT_CHILD_- RELATIONSHIP	product_definition_relationship	41		
tdp_element_parent_to_tdp_- element_child_relationship to element_identification (child)	PATH			product_definition_relationship product_definition_relationship.relatrd_product_definition->product_definition product_definition.formation->product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product_category product_category.name='document'
tdp_element_parent_to_tdp_- element_child_relationship to element_identification (parent)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition->product_definition product_definition.formation->product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product_category product_category.name='document'
TDP_INDENTURED_ITEM	product_definition	41		
tdp_identured_item to item_- identification (top_indenture)	PATH			product_definition product_definition.formation->product_definition_formation

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
TDP_INDENTURED_TDP_-ELEMENT	product_definition	41		
tdp_indentured_tdp_element to element_identification (top_indenture)	PATH			product_definition product_definition.formation-> product_definition_formation
TIME_INTERVAL_-EFFECTIVITY	time_interval_based_effectivity	41		
time_interval_effectivity to date (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)
time_interval_effectivity to date (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
time_interval_effectivity to event (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence
time_interval_effectivity to event (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence
time_interval_effectivity to label (interval_name)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval time_interval.name
time_interval_effectivity to number_-with_units (duration)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.derivation-> measure_with_unit

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
VOLUME	measure_with_unit	41		
volume to number_with_units (a_value)	IDENTICAL MAPPING			[measure_with_unit.unit_component-> unit unit=named_unit named_unit=> volume_unit] [named_unit.dimensions-> dimensional_exponents] {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-0} {dimensional_exponents.mass_exponent=0} {dimensional_exponents.length_exponent=3} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}
WEIGHT	representation_item	43		
weight to a_real (a_value)	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value measure_value=numeric_measure

Table 8 — Mapping table for indentured_data_list (continued)

Application element	AIM element	Source	Rules	Reference path
weight to measure (unit_of_measure) #1: If weight is expressed in SI units #2: If weight is not expressed in SI units	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.unit_component-> unit unit=named_unit named_unit=> #1: (si_unit si_unit.name-> si_unit_name si_unit_name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]) #2: (conversion_based_unit conversion_based_unit.name [conversion_based_unit.conversion_factor-> measure_with_unit] [measure_with_unit.value_component-> measure_value] [measure_with_unit.unit_component-> unit unit=named_unit named_unit=> si_unit si_unit.name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]])

Table 8 — Mapping table for indentured_data_list (concluded)

Application element	AIM element	Source	Rules	Reference path
weight_derivation (derivation_method)	PATH			representation_item=> [qualified_representation_item qualified_representation_item.qualifiers[i]-> value_qualifier] [measure_representation_item<= measure_with_unit<- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifier[i]-> value_qualifier] value_qualifier=type_qualifier type_qualifier.name {measure_qualification.name='weight derivation method'} {(type_qualifier.name='calculated') (type_qualifier.name='estimated') (type_qualifier.name='weighed')}

Table 9 — Mapping table for index_list

Application Element	AIM Element	Source	Rules	Reference path
ASSOCIATED_LIST #1: If the associated list is a data_list #2: If the associated list is an indentured data list #3: If the associated list is an index list	product_definition_formation	41	24	#1:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='data list' }) #2:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='indentured data list' }) #3:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='index list' })

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
associated_list to drawing (related_to)	PATH			#1: (product_definition_formation <- product_defnition_formation_relationship.related_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='associated list'} product_definition_formation_relationship.relating_product_definition_-formation-> product_definition_formation
associated_list to list_presentation (presentation)	PATH			product_definition_formation presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set
associated_list to product_data_set (related_to)	PATH			product_defination_formation<- product_defination.formation product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_defnition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name ='product data set'}

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
DATA_LIST	product_defintion_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='data list' }
data_list to data_list_body (body) (see note 3)				
data_list to data_list_header (list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=’data list header’}
DATA_LIST_HEADER	property_definition	41		{property_definition property_definition.description=’data list header’}
data_list_header to header (common_header)	IDENTICAL MAPPING			

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
DOCUMENT_LIST	product_definition_formation	41		
DRAWING	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='drawing'}
drawing to header (heading) (see note 3)				
drawing to sheet (pages) (see note 3)				

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
INDENTURED_DATA_LIST	product_definition_formation	41	19 24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='indentured data list' }
indentured_data_list to indentured_-data_list_body (body) (see note 3)				
indentured_data_list to indentured_-data_list_header (header)	PATH			product_definition_formation<- product_definition.formation (product_defnition)(product_definition_with_associated_documents) characterized_product_definition=(product_defnition)\ (product_definition_with_associated_documents) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='indentured data list header'}

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
INDEX_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='index list' }
index_list to index_list_body (body)	PATH			product_definition_formation<- product_definition.formation product_definition {product_definition.description='index list body'}
index_list to index_list_header (list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='index list header'}

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
INDEX_LIST_BODY	product_definition	41		{product_definition product_definition.description='index list body'}
index_list_body to index_list_entry (index_list_entries)	PATH			product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='index list entry'} product_definition_relationhsip.releted_product_definition-> product_definition
index_list_body to notation (index_list_notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition {property_definition.description='index list body'} property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
index_list_body to revision (revision_history)	PATH			product_definition <product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_-formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.\ relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action => {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
INDEX_LIST_ENTRY	product_definition	41		{(product_definition product_definition.frame_of_reference-> product_definition_context) (product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference-> product_definition_context) product_definition_context=> application_context_element application_context_element..name='index list entry'}
index_list_entry to header (list_entry)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definitioncharacterized_- definition=property_definition property_definition
index_list_entry to identifier (entry_item_change_level)	PATH			(product_definition action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification for entry'} action_assignment.assigned_action-> action {action=> executed_action} action.name)

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
index_list_entry to notation (entry_notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition<- {property_definition.name='index list entry property'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item representation_item=> descriptive_representation_item
INDEX_LIST_HEADER	property_definition	41		{property_definition property_definition.description='index list header'}
index_list_header to header (common_header)	IDENTICAL MAPPING			

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_-formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_-formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_-formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification->certification

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
item_identification to change_- identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
item_identification to identifer (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.itemSi] representation representation.name='notation'}

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.ro.e-> classification_role classification_role.name='reference code'} classification_assignment.assigned_class-> group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role-> classification_role classification_role.name='type of notation'} classification_assignment.assigned_class-> group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification'))}

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly with Revision when approval of person or organization is not needed. #2: date and time associated directly with Revision when approval of person or organization is not needed. #3 when approval of person or organization is needed and the date of their approval.	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
revision to other_reference_document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
revision to person_and_organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 9 — Mapping table for index_list (continued)

Application Element	AIM Element	Source	Rules	Reference path
SINGLE_DOCUMENT_LIST	product_definition_formation	41		

Table 9 — Mapping table for index_list (concluded)

Application Element	AIM Element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a index list #2: When tdp_element is an indentured_data_list #3: When tdp_element is a data_list	#1, #2, #3: (product_definition_-formation)	41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='index list'}) #2: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='indentured data list'}) #3:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name=('data list'})

Table 10 — Mapping table for other_list

APPLICATION element	AIM element	Source	Rules	Reference path
ASSOCIATED_LIST	product_definition_formation	41		
associated_list to drawing (related_to)	PATH			#1: (product_definition_formation <- product_defnition_formation_relationship.related_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='associated list'} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation
associated_list to list_presentation (presentation)	PATH			product_definition_formation presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
associated_list to product_data_set (related_to)	PATH			product_defination_formation<- product_defination.formation product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_defnition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name ='product data set'}

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or AP 202 exchange with and AP 232 exchange	#1: (product_defintion_formation) #2: ([draughting_drawing_revision] [product_defintion_formation])	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition_formation product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'})

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
drawing to header (heading)	#1: (PATH) #2: (PATH)			#1, #2: product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='drawing header'})
drawing to sheet (pages) (see note 3)				
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name=’alternate part identification’) #4: (product_definition_formation_relationship.name=‘supplied item’)} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name=’alternate part identification’} product_relationship.related_product->product<- product_definition_formation)

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
item_identification to change_-identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.items[i] representation representation.name='notation'}

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role->classification_role classification_role.name='reference code'} classification_assignment.assigned_class->group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role->classification_role classification_role.name='type of notation'} classification_assignment.assigned_class->group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
OTHER_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='other list'}
other_list to other_list_body (list_body)	PATH			proudct_definition_formation<- product_definition.formation product_definition

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
other_list to other_list_header (list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='other list header'}
OTHER_LIST_BODY	product_definition	41		{product_definition product_definition.description='other list body'}
other_list_body to notation (notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='other list body'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='notation'} representation.items[i]->representation_item=> descriptive_representation_item

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
other_list_body to other_list_body_-table (list_body_tables)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='other list body'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation=> other_list_table_representation

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
other_list_body to revision (revision_history)	PATH			product_definition <product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.\ relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action => {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action
OTHER_LIST_BODY_TABLE	other_list_table_representation	232		other_list_table_representation<= representation

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
other_list_body_table to label (description_of_tabulation)	PATH			other_list_table_representation<= representation representation.name
other_list_body_ table to label (tabulation_column_descriptions)	PATH			other_list_table_representation<= representation representation.items[i]->representation_item {representation_item.name='column names'} representation_item=> compound_representation_item compound_representation_item.item_element-> compound_item_definition compound_item_definition=list_representation_item list_representation_item.[i]-> representation_item=> descriptive_representation_item {descriptive_representation_item.name='column heading'} descriptive_representation_item.description
other_list_body_table to tabulation_-entry (tabulation_rows)	PATH			other_list_table_representation<= representation representation.items[i]-> representation_item=> {representation_item.name='identificatoin of rows'} compound_representation_item compound_representation_item.item_element-> compound_item_definition compound_item_definition=list_representation_item list_representation_item.[i]-> representation_item=> {representation_item.name='row definition'} compound_representation_item
OTHER_LIST_HEADER	property_definition	41		{property_definition property_definition.description='other list header'}

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
other_list_header to header (common_header)	IDENTICAL MAPPING			
PRODUCT_DATA_SET	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set'}
product_data_set to drawing (related) (see note 3)				
product_data_set to product_data_set (related) (see note 3)				
product_data_set to source_file (file_configuration) (see note 3)				

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_WITH_-FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set with format header'}
product_data_set_with_format to header (product_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='product data set with format header'}
product_data_set_with_format to product_presentation (presentation_of_product_data_set) (see note 3)				

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_WITH_-SHADING	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set with shading header'}
product_data_set_with_shading to header_configuration_with_-element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set with shading header'}
product_data_set_with_shading to shaded_shape_model (a_shaded_-model) (see note 3)				

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_- WITHOUT_FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set without format header'}
product_data_set_without_format to header_configuration_with_- element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='product data set without format header'}
product_data_set_without_format to shape_model (a_model) (see note 3)				

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification'))}

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly with Revision when approval of person or organization is not needed. #2: date and time associated directly with Revision when approval of person or organization is not needed. #3 when approval of person or organization is needed and the date of their approval.	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
revision to other_reference_- document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
revision to person_and_organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition_formation product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
TABULATION_ENTRY	list_representation_item	43		
tabulation_entry to text (column_information)	PATH			compound_representaion_item compound_representation_item.item_element->compound_item_definition compound_item_definition=list_representation_item list_representation_item list_representaton_item[i]->representation_item=> {representation_item.name='column entry for row'} descriptive_representation_item descriptive_representation_item.description

Table 10 — Mapping table for other_list (continued)

APPLICATION element	AIM element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a drawing defined within this document #2: When tdp_element is a product_-data_set #3: When tdp_element is a reference_document #4: When tdp_element is an other_-list #5: When tdp_element is a drawing or a product_data_set and is defined externally to this document #6: When tdp_element is a drawing defined by AP 202	#1,#2,#3,#4,#5: (product_definition_-formation) #6: (<draughting_drawing_revision> <product_definition_formation>)	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category {product_category.name='document} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='drawing'}) #2: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category (product_category.name='document'} **1**product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='product data set' }) #3:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- product_category.name='reference document' }) #4:({product_definition_formation

Table 10 — Mapping table for other_list (concluded)

APPLICATION element	AIM element	Source	Rules	Reference path
				product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='other list' }) #5:({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {(product_category.name='drawing') (product_category.name='product data set') }) #6:(<draughting_drawing_revision> {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='drawing' }

Table 11 — Mapping table for parts_list

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_IDENTIFICATION_-OF_ITEM	product_definition_formation	41		
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ALTERNATE_ITEM #1: If alternate is a substitute with no concern for which is used in an assembly context #2: If alternate is an alternate item with concern for which is used in an assembly context #3: If alternate is an alternate item with a one for one substitution and allows for two way substitution in a particular assembly context	product	41		#1: ({product<- alternate_product_relationship.alternate}) #2: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute}) #3: ({product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to an_integer (preference_order) #1: (see note 2) #3: (see note 2)	#2: (PATH)			#2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute assembly_component_usage_substitute_with_ranking assembly_component_usage_substitute_with_ranking.ranking)
alternate_item to item (interchange_item) #2: (see note 2) #3: (see note 2)	PATH			product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition
alternate_item to item_usage (interchange_item) #1: (see note 2) #3: (see note 2)	PATH			#2: product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- (product_definition_relationship.related_product_definition product_definition_relationship)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to label (type_of_alternate)	PATH			#1: (product<- alternate_product_relationship.alternate alternate_product_relationship alternate_product_relationship.name) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute assembly_component_usage_substitute.name) #3: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product_definition_substitute.name)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item to text (usage_conditions) #4:Provides document or portion of document to describe occurrence relationship #5 Multiple usage_conditions can be captured in one text string and parsed, if needed, based on an exchange partner business agreement.	#1: (PATH) #2: (PATH) #3: (PATH)			#1: ({<product<- alternate_product_relationship.basis> <product<- alternate_product_relationship.description> <alternate_product_relationship document_reference_item=alternate_product_relationship document_reference_item<- (applied_document_reference.items[i] applied_document_reference applied_document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document>}) #2: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_relationship.related_product_definition product_definition_relationship=> product_definition_usage=> assembly_component_usage<- assembly_component_usage_substitute.substitute assembly_component_usage_substitute #4: <document_reference_item=assembly_component_usage_substitute document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				document> assembly_component_usage_substitute.definition) #3: (product<- product_definition_formation.of_product product_definition_formation<- product_definition.formation product_definition<- product_definition_substitute.substitute_definition product_definition_substitute product_definition_substitute.description)
ASSEMBLY_RELATIONSHIP	(assembly_component_usage) (next_assembly_usage_occurrence) (quantified_assembly_component_usage) (specified_higher_usage_occurrence) (promissory_usage_occurrence)	44 44 44 44 44		{product_definition_relationship product_definition_relationship.relating_product_defintion-> [product_definition product_definition.frame_of_reference-> product_definition_context<= application_context_element application_context_element.name='part definition'] [product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference-> product_definition_context<= application_context_element application_context_element.name='assembly definition']}
ASSOCIATED_LIST #1: If the associated list is a parts_-list #2: If the associated list is a data_list #3: If the associated list is an other_list #4: If the associated list is an indentured data list #5: If the associated list is an index list	product_definition_formation	41		product_definition_formation product_defintion_formation.of_product-> product<- product_related_prodcut_category.products[i] product_related_product_category<= product_category #1:({product_category.name='parts list'}) #2:({product_category.name='data list'}) #3:({product_category.name='other list'}) #4:({product_category.name='indentured data list'}) #5:({product_category.name='index list'})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
associated_list to drawing (related_to)	PATH			#1: (product_definition_formation <- product_defnition_formation_relationship.\ related_product_defnition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='associated list'} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation
associated_list to list_presentation (presentation)	PATH			product_definition_formation presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set
associated_list to product_data_set (related_to)	PATH			product_defination_formation<- product_defination.formation product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_defnition.formation-> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category product_category.name ='product data set'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
DATE_EFFECTIVITY	dated_effectivity	41		
date_effectivity to date (end_date)	PATH			dated_effectivity dated_effectivity.effectivity_end_date-> date_and_time
date_effectivity to date (start_date)	PATH			dated_effectivity dated_effectivity.effectivity_start_date-> date_and_time

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or AP 202 exchange with and AP 232 exchange	#1,#2: (product_definition_formation) #3: ([draughting_drawing_revision] [product_definition_formation])	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to header (heading) #3: (see note 2)	#1: (PATH) #2: (PATH)			#1,#2: (product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='drawing header'})
drawing to sheet (pages) (see note 3)				
DRAWING_SUFFIX_NUMBER_- COMBINATION	product	41		
drawing_suffix_number_- combination to element_- identification (drawing_number)	PATH	41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=\ 'drawing suffix number combination'} product_relationship.\ related_product-> product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category.name='drawing'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_suffix_number_- combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name=\ 'drawing suffix number combination'}
EFFECTIVITY #1: When the effectivity is for a part and the next higher part is known #2: When the effectivity is for a part and the next higher part is not known	#1: (<configuration_effectivity> <(serial_numbered_effectivity) (dated_effectivity) (lot_effectivity)>) #2:[applied_effectivity_assignment] [applied_effectivity_context_assignment]	41 41 41 41 232 232		#1: ({configuration_effectivity<= product_definition_effectivity product_definition_effectivity.usage->product_definition_relationship product_definition_relationship.description='item usage'}) #2: ({applied_effectivity_assignment<= effectivity_assignment<- effectivity_context_assignment.assigned_effectivity_assignment effectivity_context_assignment [effectivity_context_assignment.role->effectivity_context_role effectivity_context_role.name='context document of effectivity'] [effectivity_context_assignment=> applied_effectivity_context_assignment applied_effectivity_context_assignment.items[i]->effectivity_context_item effectivity_context_item=product_definition_formation product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<= product.category product_category.name='document']})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to approval (effectivity_approvals)	#1: (PATH) #2: (PATH)			#1: (configuration_effectivity approval_item=configuration_effectivity approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) #2: (applied_effectivity_assignment approval_item=applied_effectivity_assignment approval_item<- applied_approval_assignment.items[i] applied_approval_assignment) <applied_approval_assignment<= approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role>
effectivity to label (name)	PATH			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.named_item name_attribute name_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity<- name_attribute_select=effectivity name_attribute_select<- name_attrubute.described_item name_attribute name_attribute.attribute_value)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to label (work_activity)	PATH			configuration_effectivity action_item=configuration_effectivity action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.name
effectivity to product_configuration (product)	#1: (PATH) #2: (PATH)			#1: (configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity-> effectivity=> product_definition_effectivity=> configuration_effectivity configuration_effectivity.configuration-> configuration_design configuration_design.configuration-> configuration_item)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to text (description)	#1: (PATH) #2: (PATH)			#1: (configuration_effectivity<= product_definition_effectivity<= effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attribute.described_item description_attribute description_attribute.attribute_value) #2: (applied_effectivity_assignment<= effectivity_assignment effectivity_assignment.assigned_effectivity->effectivity<- description_attribute_select=effectivity description_attribute_select<- description_attribute.described_item description_attribute description_attribute.attribute_value)
EXTERNAL_LIBRARY_-REFERENCE #1: If external_library_reference is an element in a class source select, that is Reference by a general_-classification as the classification_-source. #2: If external_library_reference is an element in a property_source _-select, that is reference by a property as the property_source.	#1: (externally_defined_class) #2: (externally_definied_general_-property)	232 232		#1: (externally_defined_class<= [ckass<= group][externally_defined_item]) #2:(externally_defined_general_property<= [general_property] [externally_defined_item])
external_library_reference to identifier (external_id)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.item_id

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
external_library_reference to identifier (library_type)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source->external_sourrce external_sourrce.id {external_source.source_id->source_item source_item=identifier identifier
external_library_reference to text (description)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source->externnal_source external_sourcee.description
EVENT	event_occurrence	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
event to date (actual_date) #1: Just date #2: date and time	PATH			event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name='actual date'} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name='actual date'} date_assignment.assigned_date_and_time-> date_and_time)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
event to date (planned_date)	PATH			event_occurrence #1: (date_item=event_occurrence date_item<- applied_date_assignment.item[i] applied_date_assignment=> date_assignment {date_assignment.role-> date_role date_role.name='planned date'} date_assignment.assigned_date-> date) #2: (date_and_time_item=event_occurrence date_and_time_item<- applied_date_and_time_assignment.item[i] applied_date_and_time_assignment=> date_and_time_assignment {date_and_time_assginment.role-> date_time_role date_time_role.name='planned date'} date_assignment.assigned_date_and_time-> date_and_time)
event to identifier (id)	event_occurrence.id	41		
event to label (name)	event_occurrence.name	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
event to person_and_organization (responsible_person_organization)	PATH			event_occurrence (person_and_organization_item=event_occurrence person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role person_and_organization_role.name='identifier'} person_and_organization_assignment.assigned_person_and_organization-> person_and_organization) (organization_item=event_occurrence organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role organization_role.name='system destination'} organization_assignment.assigned_organization-> organization)
event to project (assignment)	PATH			event_occurrenc<- event_occurrence_assignment.assigned_event_occurrence event_occurrence_assignment<= applied_event_occurrence_assignment applied_event_occurrence_assignment.items[i]-> event_occurrence_item event_occurrence_item=organizational_project organizational_project
event to text (description)	event_occurrence.description	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
FILE	document_file	232		{document_file<= document<- document_representation_type.representation_types document_representation_type (document_representation_type.name='digital') (document_representation_type.name='physical')}
file to change_identification (change_status)	PATH			document_file action_item=document_file action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> executed_action

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file to content_property (file_content_property)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name=document content'} {representation.context_of_items-> representation_context representation_context.context_type='document parameters'}
file to distribution_notice (distribution_authorizations)	PATH			document_file approval_item=document_file approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name=distribution notice'} approval_assignment.assigned_approval-> approval

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file to file_format (context_file_format)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'}
file to identifier (context_file_name) #1: provides alias for file identification	PATH			document_file<= document document.id #1<document _file identification_item=document_file identification_item<- applied_identification_assignment.items[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}>

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file to identifier (native_format_file_name)	PATH			(document_file<= document<- document_relationship.related_document document_relationship {document_relationship.name='translation'} document_relationship.relating_document-> document {document=> document_file} document.id
file to label (file_type_content)	PATH			document_file<= document document.kind-> document_type document_type.product_data_type
file to notation (file_note)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file to person_organization (assigned_to) #1 person and orgnaization #2: just organization	PATH			#1: (document_file persion_and_organization_item=document_file person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=persion_and_organization person_organization_select) #2: (document_file organization_item=document_file organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
file to security_classification (security_identification)	PATH			document_file security_classification_item=document_file security_classification_item<- applied_security_classification_assignment.items[i] applied_security_classification_assignment<= security_classification_assignment <security_classification_assignment role_select=security_classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role> security_classification_assignment.assigned_security_classification-> security_classification

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file to size_characteristics (size)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name=‘document property’} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
file to system_declaration (source_system)	PATH			document_file characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name=‘document property’} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {(representation.name='source system declaration’) (representation.name=‘document creation’)}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_FORMAT	representation	43		{representation.name='document format'}
file_format to date (release_date)	PATH			representation (date_item=representation date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment date_assignment.assigned_date->date {date_assignment.role->date_role date_role.name='release date'}) (date_and_time_item=representation date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role->date_time_role date_time_role.name='release date'} date_and_time_assignment.assigned_date_and_time->date_and_time)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (change_level)	PATH			representation action_item=representation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action {action=> executed_action} action.name {action.chosen_method-> action_method action_method.name='change level'}
file_format to identifier (format_code)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='character code'} descriptive_representation_item descriptive_representation_item.description

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (revision_level)	PATH			representation action_item=representation acton_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action action.name {action=> executed_action) {action.chosen_method-> action_method action_method.name='revision level'}
file_format to text (format_standard)	PATH			representation {representation.name='document format'} representation.items[i]-> representation_item=> {representation_item.name='data format'} descriptive_representation_item descriptive_representation_item.description

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_RELATIONSHIP	document_relationship	41		
file_relationship to file (related_file)	PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to file (relating_file)	PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to label (name)	PATH			document_relationship document_relationship.name
file_relationship to text (description)	PATH			document_relationship document_relationship.description

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
FOREIGN_DEFINED_ITEM #1: Related item_list is an assembly where the foreign_defined_item is a component. Component is identified on assembly drawing but not defined there. #2: Related item_list is an non assembly where the foreign_-defined_item is just a member of the group of parts listed for some business reason. (Example - list of parts for a kit)	#1,#2: (product_definition_relationship)	41		#1: ({product_definition_relationship product_definition_relationship.related_product_definition-> product_definition<- product_definition_relationship.related_product_definition product_definition_relationship [product_definition_relationship.name='foreign defined'] [product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category<- product_category.name='document' product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category (product_category.name) (product_category.name='drawing')]}) #2: ({product_definition_relationship product_definition_relationship.name='foreign defined'})
GENERAL_ITEM_DEFINITION_-RELATIONSHIP #1 Common mapping for replacement or derivation of item definition. #2 Optional mapping for derivation to an item definition to be compatible with ISO 10303-203	#1 (product_definition_relationship) #2 (design_make_from_relationship)	41 232		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
general_item_definition_relationship to item (base_item_definition)	#1: (PATH) #2: (PATH)			#1: (product_definition_relationship product_definition_relationship.related_product_definition-> product_definition) #2 (design_make_from_relationship<= product_definition_usage<= product_definition_relationship product_definition_relationship.related_product_definition-> product_definition)
general_item_definition_relationship to label (relation_type)	#1: (PATH) #2: (NOT APPLICABLE)			#1: (product_definition_relationship product_definition_relationship.name {(product_definition_relationship.name) (product_definition_relationship.name=\ 'replacement_definition') (product_definition_relationship.name=\ 'derivation')}
general_item_definition_relationship to text (description)	product_definition_-relationship.description	41		
INDEPENDENT_PROPERTY	general_property	41		
independent_property to external_-library_reference (property_source)	IDENTICAL_MAPPING			{general_property=> externally_defined_general_property}
independent_property to identifier (id)	general_property.id	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property to measure (allowed_unit) #1: If the unit has a commonly used descriptor. #2: If the unit descriptor is constructed of more than one unit name.	PATH			general_property represented_definition=general_property represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='allowed units'} representation.context_of_item-> representation_context=> global_unit_assigned_context.units[i]-> unit #1: (unit=named_unit named_unit) #2: (unit=derived_unit derived_unit)
independent_property to text (description)	general_property.description	41		
INDEPENDENT_PROPERTY_-RELATIONSHIP	general_property_relationship	41		
independent_property_relationshp to text (description)	general_property_relationship.description	41		
independent_property_relationship to independent_property (related_property)	PATH			general_property_relationship general_property_relationship.related_property-> general_property

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property_relationship to independent_property (relating_property)	PATH			general_property_relationship general_property_relationship.relating_property->general_property
independent_property_relationship to label (relation_type)	general_property_relationship.name	41		
INDEPENDENT_PROPERTY_-USAGE	general_property_association	41		
independent_property_usage to independent_property (assigned_property)	PATH			general_property_association general_property_association.base_definition->general_property
independent_property_usage to property_assignment (assigned_to)	PATH			general_property_association general_property_association.derived_definition->derived_property_select derived_property_select=property_definition property_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification->certification

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to change_- identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_-suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product->product

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.product[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_-information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ITEM_LIST #1: If the item_list is for a part or assembly and does not identify the drawing or document that the part or assembly is represented on. #2: If the item_list is for a part or assembly and the item_list_drawing is identified for the part or assembly. #3: If the item_list is for an item_-list_drawing and no part or assembly is identified on the item_list_-drawing.	product_definition	41		#1,#2,#3: {product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference-> product_definition_context<= application_context_element application_context_element.name='item_list'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_list to component_list_type_- enumeration (type_of_component_list)	#1: (PATH) #2: (PATH) #3: (PATH)			product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference-> product_definition_context<= application_context_element application_context_element.name #1: ({(application_context_element.name) (application_context_element.name=\ 'assembly defined on drawing component list') (application_context_element.name='made from component list of one') (application_context_element.name='synthetic part number component list') (application_context_element.name='net change list')}) #2: ({(aplication_context_element.name) (appliciation_context_element.name=\ 'assembly defined on drawing component list') (application_context_element.name='made from component list of one') (application_context_element.name='synthetic part number component list') (application_context_element.name='installation drawing component list') (application_context_element.name=\ 'item not defined on next higher assembly component list') (application_context_element.name='net change list')}) #3: ({(application_context_element.name) (application_context_element.name='installation drawing component list') (application_context_element.name=\ 'item not defined on next higher assembly component list') (application_context_element.name='net change list')})
item_list to item_identification (list_for)	#1: (PATH) #3, #2: (NOT APPLICABLE)			#1: (product_definition<- product_definition_formation.of_product product_definition_formation)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_list to item_list_drawing (list_for)	#1: (NOT APPLICABLE) #2: (PATH) #3: (PATH)			#2: (product_definition<- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.description='parts on drawing'} product_definition_relationship.relating_product_definition->product_definition product_definition.formation->product_definition_formation) #3: (product_definition product_definition.formation->product_definition_formation)
item_list to item_usage (component_of)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2,#3: (product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {(product_definition_relationship.description) (product_definition_relationship.description=('item usage'))})
item_list to text (type_of_component_list)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2,#3: product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference->product_definition_context<= application_context_element application_context_element.name
ITEM_LIST_DRAWING	(product_definition_formation)	41		{product_definition_formation.of_product->product<- product_related_product_category.products product_related_product_category<= product_category (product_category.name='document') (product_category.name='sheet')}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_list_drawing to element_-identification (drawing_number)	IDENTICAL MAPPING			
item_list_drawing to item_-identification (assembly_item_list_applies_to)	PATH			product_definition_formation<- product_definition.formation product_definition product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.description='parts on drawing'} product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment=> organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name=’code administator’} organization_assignment.assigned_organization-> organization
item_type to label (item_code)	PATH			product_category product_category.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description
ITEM_USAGE #1: If the item_list (that is using the item_usage) is for a part or assembly. #2: If the item_list (that is using the item_usage) is for an item_list_-drawing and no part or assembly is identified on the item_list_drawing (example-items in a kit). #3: An Alternate_item is using the Item_usage	#1, #2, #3: (product_definition_-relationship) (Note: Can be all subtypes of product_-definition_relationship also)	41		#1,#2: ({product_definition_relationship (product_definition_relationship.description) (product_definition_relationship.description='item usage')}) #3: ({product_definition_relationship (product_definition_relationship.description) (product_definition_relationship.description='substitution')})
item_usage to effectivity (effective_on)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2, #3: (product_definition_relationship<- product_definition_effectivity.usage product_definition_effectivity=> configuration_effectivity)
item_usage to item (being_defined_for)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2, #3: (product_definition_relationship product_definition_relationship.related_product_definition> product_definition {product_definition<- product_definition_context_association.definition product_definition_context_association product_definition_context_association.frame_of_reference product_definition_context<= application_context_element application_context_element.name='item in list'})
item_usage to list_item_usage (list_item)	#1: (IDENTICAL MAPPING) #2: (IDENTICAL MAPPING) #3: (IDENTICAL MAPPING)			

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_usage to quantity (quantity_used_in_next_higher_-assembly)	#1: (PATH) #2: (PATH) #3: (PATH)			#1, #3: (product_definition_relationship=> product_definition_usage=> assembly_component_usage=> ([next_assembly_usage_occurrence] [assembly_component_usage=> quantified_assembly_component_usage quantified_assembly_component_usage.quantity-> measure_with_unit]) (assembly_component_usage=> quantified_assembly_component_usage quantified_assembly_component_usage.quantity-> measure_with_unit)) #2: (product_definition_relationship characterized_product_definition=product_definition_relationship characterized_definition=characterized_product_definition characterized_definition<- property_definition {property_definition.name=\ 'component quantity'}<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='component quantity representation'} representation.items[i]-> representation_item=> measure_representation_item<= measure_with_unit)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
item_usage to retrofit_usage (retrofit)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2, #3: (product_definition_relationship action_item=product_definition_relationship action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='retrofit usage'}
item_usage to special_condition (special_conditions)	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2, #3: (product_definition_relationship product_definition_relationship.related-> product_definition characterized_product_definition=product_definition characterized_product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='item usage properties'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='special condition'} representation.items[i]-> representation_item=> descriptive_representation_item)

--`,,`-`-`,,`,,`,`,,`---

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
LIST_ITEM_USAGE #1: If the item_list (that is using the item_usage) is for a part or assembly. #2: If the item_list (that is using the item_usage) is for an item_list_-drawing and no part or assembly is identified on the item_list_drawing (example - items in a kit).	product_definition_relationship	41		
list_item_usage to approval (approvals)	PATH			product_definition_relationship approval_item=product_definition_relationship approval_item<- applied_approval_assignment.items applied_approval_assignment

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
list_item_usage to identifier (drawing_zones)	PATH			#1: (product_definition_relationship product_definition_relationship.related_product_definition-> product_definition<- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.relating_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {(product_category.name='document') (product_category.name='sheet')} product_category_relationship.category product_category_relationship product_category_relationship.subcategory-> product_category product_category.name='drawing'} characterized_product_definition=product_definition_relationship characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition<- {property_definition.name='item usage properties'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='drawing zones'} representation.items[i]-> representation_item=> {representation_item.name='drawing zone'} descriptive_representation_item descriptive_representation_item.description) #2: (product_definition_relationship characterized_product_definition=product_definition_relationship characterized_definition=characterized_product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition<- property_definition.definition property_definition<- {property_definition.name='item usage properties'} property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='drawing_zones'} representation.items[i]-> representation_item=> {representation_item.name='drawing zone'} descriptive_representation_item descriptive_representation_item.description)
list_item_usage to identifier (entry_item_change_level)	PATH			product_definition_relationship action_item=product_definition_relationship action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification for entry'} action_assignment.assigned_action-> action action.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
list_item_usage to identifier (find_number) #3: Usage find number for a component in an assembly (item) with possible effectivity driving uniqueness. #4: Document find number for a component in an assembly (item). #5: Document find number for a component in an item_list and the item_list is not a part assembly, but the document that formulates the collection of parts. #6: Full document #7 Portion of document	PATH			#3: (product_definition_relationship product_definition_relationship.name) #4: ((product_definition_relationship product_definition_relationship.related_product_definition-> product_definition<- product_definition_relationship.related_product_definition product_definition_relationship product_definition_relationship.name) (product_definition_relationship document_reference_item=product_definition_relationship document_reference_item<- #6: (applied_document_reference.items[i] applied_document_reference identificaiton_item=applied_document_reference identification_item<- applied _identification_assignment.items[i] applied_identification_assignment<= identificaiton_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.label='document find number'}) #7: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment identificaiton_item=applied_document_usage_constraint_assignment identification_item<- applied _identification_assignment.items[i] applied_identification_assignment<= identificaiton_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.label='document find number'})) (product_definition_relationship characterized_property_definition=product_definition_relationship characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition representation_definition=property_definition representation_definition<- property_definition_representation.definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.used_representation-> representation representation.item[i]-> representation_item<= descriptive_representation_item descriptive_reprresentation_item.description {descriptive_representation_item.name='document find number'} <descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- applied_document_reference.items[i] applied_document_reference applied_document_reference.assigned_document-> document>)) #5: (product_definition_relationship product_definition_relationship.name)
list_item_usage to identifier (reference_designators)	PATH			product_definition_relationship=> product_definition_usage=> assembly_component_usage (assembly_component_usage) (assembly_component_usage=> {make_from_usage_option_with_reference_designator}) assembly_component_usage.reference_designator
list_item_usage to label (item_type)	PATH			product_definition_relationship product_definition_relationship.related_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product._related_product_category.products[i] product_related_product_category<= product_category product_category.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
list_item_usage to notation (entry_notes)	PATH			product_definition_relationship characterized_product_definition=product_definition_relationship characterized_definition=characterized_product_definition characterized_definition<- property_definition {property_definition.name='item usage properties'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item
list_item_usage to security_-classification (security_information)	PATH			product_definition_relationship security_classification_item=product_definition_relationship security_classification_item<- applied_security_classification_assignment.items[i] applied_security_classification_assignment<= security_classification_assignment <security_classification_assignment role_select=security_classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role> security_classification_assignment.assigned_security_classification-> security_classification

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
LOCALLY_DEFINED_ITEM #1: Related item_list is an assembly where the Locally_defined_item is a component. Component is identified on assembly drawing but not defined there. The component part is defined on another drawing. #2: Related item_list is an assembly where the Locally_defined_item is a component. Component is identified and defined on the assembly drawing. #3: Related item_list is an non assembly where the Locally_-defined_item is just a member of the group of parts listed for some business reason. (Example - list of parts for a kit) The Locally_defined_item is defined on an other document.	#1,#2,#3: (product_definition_-relationship)	41		#1,#2,#3: ({product_definition_relationship product_definition_relationship.related_product_definition-> product_definition<- product_definition_relationship.related_product_definition product_definition_relationship [product_definition_relationship.relating_product_definition-> product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.subcategory-> product_category (product_category.name) (product_category.name='drawing')] [(product_definition_relationship.name) (product_definition_relationship.name='locally defined')]})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
LOCATION_INSTANCE	axis2_placement	42		
location_instance to a_real (scale)	PATH			axis2_placement (axis2_placement=axis2_placement_2d axis2_placement_2d<=) (axis2_placement=axis2_placement_3d axis2_placement_3d<=) placement<= geometric_representation_item<= representation_item<- representation.item[i] representation representation.item[i]-> representation_item=> geometric_representation_item-> cartesian_transformation_operator (cartesian_transformation_operator) (cartesian_transformation_operator=> (cartesian_transformation_operator_2d) (cartesian_transformation_operator_3d)) cartesian_transformation_operator.scale

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
location_instance to file (geometry_definition)	PATH			axis2_placement (axis2_placement=axis2_placement_2d axis2_placement_2d<=) (axis2_placement=axis2_placement_3d axis2_placement_3d<=) placement<= geometric_representation_item<= representation_item<- representation.item representation<- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> represented_definition represented_definition=property_definition property_definition property_definition.definition-> characterized_definition characterized_definition=characterized_object characterized_object=> document_file
location_instance to geometry (geometry_definition)	PATH			axis2_placement (axis2_placement=axis2_placement_2d axis2_placement_2d<=) (axis2_placement=axis2_placement_3d axis2_placement_3d<=) placement<= geometric_representation_item

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
location_instance to identifier (id)	PATH			axis2_placement (axis2_placement=axis2_placement_2d axis2_placement_2d<= placement<= geometric_representation_item<= representation_item representation_item.name) (axis2_placement=axis2_placement_3d axis2_placement_2d<= placement<= geometric_representation_item<= representation_item<- representation.item representation representation.id)
location_instance to transformation (transformation_definition)	IDENTICAL MAPPING			
LOT_EFFECTIVITY	lot_effectivity	41		
lot_effectivity to a_number (lot_size)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.value_component-> measure_value
lot_effectivity to identifier (lot_identification)	PATH			lot_effectivity lot_effectivity.effectivity_lot_id

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
lot_effectivity to measure (lot_unit_of_measure)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size-> measure_with_unit measure_with_unit.unit_component-> unit
MADE_FROM_STOCK_-MATERIAL	(make_from_usage_option) (make_from_usage_option _with_reference_designator)	44 232		{make_from_usage_option<= product_definition_usage<= product_definition_relationship product_definition_relationship.name='made from stock'} {make_from_usage_option_with_reference_designator<= [make_from_usage_option] [assembly_component_usage]}
made_from_stock_material to a_-number (quantity_of_item_made_from_-stock)	PATH			make_from_usage_option make_from_usage_option.quantity-> measure_with_unit

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
made_from_stock_material to reference_document_usage (specified_by)	PATH			make_from_usage_option document_reference_item=make_from_usage_option document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='specifies made from stock material'} document_reference_item=document_reference document_reference_item<- applied_document_usage_constraint_assignment.items[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment {document_usage_constraint_assignment.role-> document_usage_role document_usage_role.name='specifies made from stock material'} document_usage_constraint_assignment.assigned_document_usage-> document_usage_constraint
made_from_stock_material to stock_material (stock)	PATH			make_from_usage_option<= product_definition_usage<= product_definition_relationship product_definition_relationship.related_product_definition> product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- material_designation.definitions material_designation

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
made_from_stock_material to stock_size (size)	PATH			make_from_usage_option<= product_definition_usage<= product_definition_relationship characterized_product_definition=product_definition_relationship characterized_definition=charactrized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='size'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='stock size parameters'}
MAKE_FROM_RELATIONSHIP	(make_from_usage_option) (make_from_usage_option_with_-refernce_designator)	41 232		({make_from_usage_option<= product_definition_usage<= product_definition_relationship) ({make_from_usage_option_with_reference_designator<= [make_from_usage_option] [assembly_component_usage]})
make_from_relationship to number_-with_unit (quantity_of_item_made_from_-other_item)	PATH			make_from_usage_option make_from_usage_option.quantity-> measure_with_unit
make_from_relationship to item (item_that_is_acting_as_stock)	PATH			make_from_usage_option<= product_definition_usage<= product_definition_relationship product_definition_relationship.related_product_definition-> product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
MEASURE	named_unit	41		
NOTATION	descriptive_representation_item	45		{descriptive_representation_item<= representation_item<- representation.items[i] representation representation.name='notation'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to element_identification (referenced_document_in_note) #1: reference full document #2: reference portion of document	PATH			descriptive_representation_item document_reference_item=descriptive_representation_item document_reference_item<- #1: (applied_document_reference.items[i] applied_document_reference<= document_reference <document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name> document_reference.assigned_document->) #2: (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to identifier (reference_code)	PATH			descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role-> classification_role classification_role.name='reference code'} classification_assignment.assigned_class-> group group.name {group=> class}
notation to label (type_of_notation)	PATH		23	descriptive_representation_item classification_item=descriptive_representation_item classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= classification_assignment {classification_assignment classification_assignment.role classification_role classification_role.name='type of notation'} classification_assignment.assigned_class-> group group.name {group=> class_system}
notation to label (note_title)	PATH			descriptive_representation_item descriptive_representation_item.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
notation to text (note)	PATH			descriptive_representation_item descriptive_representation_item.description
notation to text (note_parameter)	PATH			descriptive_representation_item<= representation_item<- representation_item_relationship.relating_representation_item representation_item_relationship {representation_relationship.name='note parameter'} representation_item_relationship.related_representation_item-> representation_item=> (descriptive_representation_item) (measure_representation_item) (value_representation_item)
NUMBER_WITH_UNITS	measure_with_unit	41		
number_with_units to a_real (value_of)	PATH			measure_with_unit measure_with_unit.value_component measure_value
number_with_units to measure (units_of)	IDENTICAL MAPPING			measure_with_unit measure_with_unit.unit_component unit
PART_OCCURRENCE_IN_- ASSEMBLY	assembly_component_usage	44		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
part_occurrence_in_assembly to location_instance (location) #1: Placement information based on part's representations being referenced to an additional representation. #2: Allows mapping of occurrence location through assembly relationship. #3: Allows mapping of occurrence location through occurrence instance in the case when part occurrence is located in the coordinate system of a generic origin and not in the coordinate system of the parent assembly. #4: Optional relation between a part's occurrence and its definitional view. #5: Placement information based on part's representation being mapped into additional representation. #6: The component (related) is used only once within the assembly (relating). #7: The component is used multiple times with the assembly. #8:Provides document or portion of document to describe occurrence relationship	PATH			#1:(assembly_component_usage<= product_definition_usage<= product_definition_relationship [#2: (characterized_product_definition=product_definition_relationship) #3: (product_definition_relationship<- product_definition_occurrence_relationship.occurrence_usage product_definition_occurrence_relationship #8: <document_reference_item=product_definition_occurrence_relationship document_reference_item<- (applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) (applied_document_usage_constraint_assignment.item[i] applied_document_usage_constraint_assignment<= document_usage_constraint_assignment document_usage_contraint_assignment.assigned_document_usage-> document_usage_constraint document_usage_constraint.source->) document> product_definition_occurrence_relationship.occurrence-> product_definition #4: <product_definition<- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name='definition usage'} product_definition_relationship.relating_product_definition-> product_definition> characterized_product_definition=product_definition) characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition=> product_definition_shape<- context_dependent_shape_representation.represented_product_relation context_dependent_shape_representation context_dependent_shape_representation.representation_relation-> shape_representation_relationship<=] [product_definition_relationship.relating_product_definition-> product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.defintion property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} representation_relationship.rep_2] [product_definition_relationship.related_product_definition-> product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_representation_definition} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} representation_relationship.rep_1] representation_relationship=> representation_relationship_with_transformation

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_relationship_with_transformation.transformation_operator-> transformation transformation=item_defined_transformation item_defined_transfromation item_defined_transformation.transform_item_1-> representation_item=> placement=> geometric_representation_item=> placement=> axis2_placement] [item_defined_transformation.transform_item_2-> representation_item=> geometric_representation_item=> placement=> axis2_placement]) #5: (assembly_component_usage<= product_definition_usage<= [product_definition_relationship product_definition_relationship.relating_product_definition-> product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation {representation=> shape_representation} representation.item[i]-> representation_item=>] [product_definition_relationship

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.related_product_definition-> product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation<- {representation=> shape_representation} representation_map.mapped_representation representation_map<- mapped_item.mapping_source] [#6: (product_definition_relationship) #7: (product_definition_relationship characterized_product_definition=product_definition_relationship characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition=> product_definition_shape} represented_definition=property_definition represented_defintion<- property_definition_representation.definition property_definition_representation {property_definition_representation=> shape_definition_representation} property_definition_representation.used_representation-> representation {representation=>

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation} representation.items[i]->representation_item=>)] mapped_item mapped_item.mapping_target->representation_item=> geometric_representation_item=> placement=> axis2_placement)
PARTS_LIST	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.producst[i] product_related_product_category<=product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category product_category.name='parts list'} <{product_definition_formation<- document_product_association.related.product document_product_association {document_product_association=> document_product_equivalence} document_product_association.relating_document->document document.kind->document_type document_type.product_data_type=' 'configuration controlled document version'}>
parts_list to parts_list_body (list_body)	PATH			product_definition_formation<- product_definition.formation product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
parts_list to parts_list_header (list_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='parts list header'}
PARTS_LIST_BODY	product_definition	41		{product_definition product_definition.description='parts list body'}
parts_list_body to item_list (parts_tabulations) #1: If the item_list is for a part or assembly and does not identify the drawing or document that the part or assembly is represented on. #2: If the item_list is for a part or assembly and the item_list_drawing is identified for the part or assembly. #3: If the item_list is for an item_-list_drawing and no part or assembly is identified on the item_list_-drawing.	#1: (PATH) #2: (PATH) #3: (PATH)			#1,#2,#3: (product_definition<- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name='parts tabulation'} product_definition_relationship.related_product_definition->product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
parts_list_body to notation (list_notes)	PATH			product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='parts list body property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
parts_list_body to revision (revision_history)	PATH			product_definition <product_definition.formation-> product_definition_formation<- product_definition_formation_relationship.related_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) (product_definition_formation_relationship.name='sequence') (product_definition_formation_relationship.name='iteration')} product_definition_formation_relationship.relating_product_definition_formation-> product_definition_formation> action_item=product_definition action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='revision history'} action_assignment.assigned_action-> action => {(action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification')} executed_action

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
parts_list_body to tdp_element_list_-item (referenced_tdp_elements)	PATH			product_definition<- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name='referenced tdp elements'} product_definition_relationship.relating_product_definition->product_definition {product_definition.formation->product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<=product_category product_category.name='document'}
PARTS_LIST_HEADER	property_definition			{property_definition property_definition.description='parts list header'}
parts_list_header to header (common_header)	IDENTICAL MAPPING			

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set'} <{product_definition_formation<- document_product_association.relating_document document_product_association[=> document_product_equivalence] document_product_association.related_document-> document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'}>
product_data_set to drawing (related) (see note 3)				
product_data_set to product_data_set (related) (see note 3)				
product_data_set to source_file (file_configuration) (see note 3)				

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_WITH_-FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set with format'} <{product_definition_formation<- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_document-> document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'})>
product_data_set_with_format to header (product_header)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='product data set with format header'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_with_format to product_presentation (presentation_of_product_data_set) (see note 3)				
PRODUCT_DATA_SET_WITH_-SHADING	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set with shading'}
product_data_set_with_shading to header_configuration_with_-element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_defintion=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set with shading header'}
product_data_set_with_shading to shaded_shape_model (a_shaded_-model) (see note 3)				

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_- WITHOUT_FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='product data set without format'} <{product_definition_formation<- document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_document-> document document.kind-> document_type document_type.product_data_type=\ 'configuration controlled document version'}>
product_data_set_without_format to header_configuration with_- element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set without format header'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_without_format to shape_model (a_model) (see note 3)				
PROJECT	organizational_project	41		
project to date (end_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role->date_role (date_role.name) (date_role.name=’actual end’) (date_role.name='planned end')} date_assignment.assigned_date->date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role->date_time_role (date_time_role.name) (date_time_role.name=’actual end’) (date_time_role.name='planned end')} date_and_time_assignment.assigned_date->date_and_time)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
project to date (start_date) #1: Just date #2: date and time	PATH			organizational_project #1: (organizational_project date_item=organizational_project date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment {date_assignment.role->date_role (date_role.name) (date_role.name='actual start') (date_role.name='planned start')} date_assignment.assigned_date->date) #2: (organizational_project date_and_time_item=organizational_project date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment.role->date_time_role (date_time_role.name) (date_time_role.name='actual start') (date_time_role.name='planned start')} date_and_time_assignment.assigned_date->date_and_time)
project to identifier (id)	organizational_project.id	41		organizational_project id_attribute_select=organizational_project id_attribute_select<- id_attribute.identified_item id_attribute id_attribute.attribute_value
project to label (name)	organizational_project.name	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
project to perrson_and_organization (participants)	PATH			organizational_project (person_and_organization_item=organizational_project person_and_organization_item<- applied_person_and_organization_assignment.item[i] applied_person_and_organization_assignment=> person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization-> person_and_organization person_organization_select=person_and_organization person_organization_select) (organization_item=organizational_project organization_item<- applied_organization_assignment.item[i] applied_organization_assignment=> organization_assignment organization_assignment.assigned_organization-> organization person_organization_select=organization person_organization_select)
project to person_and_organization (project_owner)	PATH			organizational_project organizational_project.responsible_organization[i]-> organization person_organization_select=organization person_organization_select
project to text (description)	organizational_project.description	41		
PROMISSORY_USAGE	promissory_usage_occurrence	44		
PROPERTY_ASSIGNMENT	property_definition	41		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
property_assignment to identifiier (described_element)	PATH			property_definition property_definition.definition-> characterized_definition
property_assignment to label (name)	property_definition.name	41		
property_assignment to text (description)	property_definition.description	41		
QUANTIFIED_PART_USAGE_-IN_-ASSEMBLY	next_assembly_usage_occurrence	44		
QUANTITY	measure_with_unit	41		
quantity to a_number (value_of)	PATH			measure_with_unit measure_with_unit.value_component-> measure_value measure_value=count_measure count_emasure
quantity to text (quantity_accuracy)	PATH			measure_with_unit<- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifier-> type_qualifier type_qualifier.name {(type_qualifier.name) (type_qualifier.name=‘exact’) (type_qualifier.name=‘as required’) (type_qualifier.name=‘approximate’)}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
quantity to text (value_of)	PATH			measure_with_unit measure_with_unit.value_component-> measure_value measure_value=descriptive_measure descriptive_measure
quantity to volume (value_of)	IDENTICAL MAPPING			
quantity to weight (value_of)	PATH			measure_with_unit=> measure_representation_item<= representation_item
RELATIVE_EVENT	relative_event_occurrence	41		
relative_event to event (related_event)	PATH			relative_event_occurrence relative_event_occurrence.base_event-> event_occurrence
relative_event to number_with_units (offset)	PATH			relative_event_occurrence relative_event_occurrence.offset-> time_measure_with_unit=> measure_with_units
RETROFIT_USAGE	applied_action_assignment	232		{applied_action_assignment<= action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='retrofit usage'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
retrofit_usage to notation (retrofit_description)	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action characterized_action_definition=action characterized_action_definition<- action_property.definition action_property<- action_property_representation.property action_property_representation {action_property_representation.name='retrofit notation'} action_property_representation.representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item
retrofit_usage to other_reference_- document (defining_document)	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.chosen_method-> (action_method=> action_method_with_associated_documents action_method_with_associated_documents.specification->) (action_method document_reference_item=action_method document_reference_item<- applied_document_reference.item[i] applied_document_reference=> document_reference document_reference.assigned_document->) document

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
retrofit_usage to retrofit_state (disposition_for)	PATH			applied_action_assignment<= action_assignment action_assignment.assigned_action-> action action.chosen_method-> action_method action_method.name {(action_method.name) (action_method.name='add') (action_method.name='delete') (action_method.name='modify')}
REVISION	executed_action	41		{executed_action<= action (action.description) (action.description='revision') (action.description='change') (action.description='issue') (action.description='sequence') (action.description='change identification'))}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to date (revison_date) #1: just date associated directly with Revision when approval of person or organization is not needed. #2: date and time associated directly with Revision when approval of person or organization is not needed. #3 when approval of person or organization is needed and the date of their approval.	PATH			#1: ((executed_action date_item=executed_action) (executed_action<= action date_item=action) date_item<- applied_date_assignment.items[i] applied_date_assignment<= date_assignment <date_assignment.role-> date_role date_role.name> date_assignment.assigned_date-> date) #2: ((executed_action date_and_time_item=executed_action) (executed_action<= action date_and_time_item=action) date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment <date_and_time_assignment.role-> date_time_role date_time_role.name> date_and_time_assignment.assigned_date_and_time-> date_and_time data_and_time.date_component-> date) #3: (executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<-

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
				role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- approval_date_time.dated_approval approval_date_time approval_date_time.date_time-> date_time_select (date_time_select=date) (date_time_select=date_and_time))
revision to identifier (revision_level) #1: to sequence revisions	PATH			executed_action<= action action.name #1: <executed_action<= action<- action_relationship.relating_action action_relationship {action_relationship.name='sequence'} action_relationship.related_action-> action>

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to other_reference_- document (authorizing_documents)	PATH		12	executed_action document_reference_item=executed_action document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role (object_role.name) (object_role.name='revision authorizing identification') (object_role.name='change rationale')} document_reference.assigned_document-> document<- {document.kind-> document_type.product_data_type=\ 'configuration controlled document version'} document_product_association.relating_document document_product_association {document_product_association=> document_product_equivalence} document_product_association.related_product-> product_or_formation_or_definition product_or_formation_or_definition=product_definition_formation product_definition_formation

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
revision to person_and_organization (revision_approval) #1: without person #2: with person	PATH		27	executed_action<= action approval_item=action approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment <approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role {(object_role.name) (object_role.name='revision approval') (object_role.name='release authenication')}> approval_assignment.assigned_approval-> approval<- #1: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=organization) #2: (approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization-> person_organization_select person_organization_select=person_and_organization person_and_organization person_and_organization.the_organization-> organization)
revision to text (revision_description)	PATH			executed_action<= action action.chosen_method-> action_method action_method.description

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
SEQUENCE_EFFECTIVITY	serial_numbered_effectivity	41		
sequence_effectivity to identifier (from_effectivity_id)	PATH			serial_numbered_effectivity serial_numbered_effectivity.effectivity_start_id
sequence_effectivity to identifier (thru_effectivity_id)	PATH			serial_numbered_effectiivity serial_numbered_effectivity..effectivity_end_id
sequence_effectivity to measure (quantity_unit_of_measure)	PATH		10	serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit measure_with_unit.unit_component-> unit
sequence_effectivity to quantity (total_component_quantity_in_ product_configuration)	PATH			serial_numbered_effectivity<= effectivity=> product_definition_effectivity product_definition_effectivity.usage-> product_definition_relationship=> product_definition_usage=> assembly_component_usage=> quantified_assembly_component_usage.quantity-> measure_with_unit

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition_formation product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description=’sheet content’} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description=’sheet content’} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name=‘size format') (representation_item.name=‘size format standard’)} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
SOURCE_INFORMATION_TYPE	representation	43	22	{representation representation.name='source information type'}
source_information_type to identifier (source_code)	PATH			representation representation.items[i]-> representation_item {representation_item=> descriptive_representation_item} representation_item.name
source_information_type to text (source_description)	PATH			representation representation.items[i]-> representation_item=> descriptive_representation_item descriptive_representation_item.description
source_information_type to text (type_of_coding_scheme)	PATH			representation representation.context_of_items[i]-> representation_context representation_context.context_type
SPECIAL_CONDITION	descriptive_representation_item	45		
special_condition to label (code)	PATH			descriptive_representation_item<= representation_item representation_item.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
special_condition to text (description)	PATH			descriptive_representation_item descriptive_representation_item.description
special_condition to text (type_of_coding_scheme)	PATH			descriptive_representation_item<= representation_item<- representation.items[i] representation {representation.name='special_condition'} representation.context_of_items[i]->>br>representation_context representation_context.context_type
SPECIFICATION_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product->>br>product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category->>br>product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category->>br>product_category product_category.name='specification document'}
SPECIFIED_PART_IN_-ASSEMBLY_TREE	specified_higher_usage_occurrence	44		
specified_part_in_assembly_tree to assembly_relationship (higher_assembly)	specified_higher_usage_-occurrence.upper_usage	44		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
specified_part_in_assembly_tree to quantified_part_usage_in_assembly (sub_assembly)	specified_higher_usage_-occurrence.next_usage	44		
STANDARD_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='standard document'}
STOCK_MATERIAL	material_designation	45		
stock_material to item (material)	PATH			material_designation material_designation.definitions-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
stock_material to stock_size (size)	PATH			material_designation material_designation.definitions-> characterized_definition characterized_definition=characterized_product_definition characterized_product_definition=product_definition product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='size'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='stock size parameters'}
STOCK_SIZE	representation	45		
stock_size to stock_size_- classification (catalog)	PATH			representation representation.item-> representation_item=> descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description) (descriptive_representation_item.description='bar') (descriptive_representation_item.description='plate') (descriptive_representation_item.description='sheet') (descriptive_representation_item.description='pipe') (descriptive_representation_item.description='tube') (descriptive_representation_item.description='block') (descriptive_representation_item.description='sphere') (descriptive_representation_item.description='roll')} {descriptive_representation_item.name='stock size classification'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
stock_size to stock_size_cross_-section (cross_section)	PATH			representation re;resentation.item-> representation_item=> descriptive_representation_item descriptive_representation_item.description {descriptive_representation_item.description) (descriptive_representation_item.description='round') (descriptive_representation_item.description='square') (descriptive_representation_item.description='oval') (descriptive_representation_item.description='hex') (descriptive_representation_item.description='angle') (descriptive_representation_item.description='channel') (descriptive_representation_item.description='rectangular') (descriptive_representation_item.description='T's') (descriptive_representation_item.description='I's')} {descriptive_representation_item.name='cross section'}
stock_size to stock_size_parameters (parameters_for)	PATH			representation representation.item[i]-> representation_item=> measure_representation_item
STOCK_SIZE_PARAMETERS	measure_representation_item	45		{measure_representation_item<= representation_item<- representation.items[i] representation representation.name='stock size parameters'}
stock_size_parameter to label (name_of_parameter)	PATH			measure_representation_item<= representation_item representation_item.name

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
stock_size_parameters to number_-with_units (a_length)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation_item.name='length of stock'}
stock_size_parameters to number_-with_units (diameter)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation._item.name='diameter of stock'}
stock_size_parameters to number_-with_units (height)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation_item.name='height of stock'}
stock_size_parameters to number_-with_units (numeric_parameter)	PATH			measure_representation_item<= measure_with_unit
stock_size_parameters to number_-with_units (radius)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation_item.name='radius of stock'}
stock_size_parameters to number_-with_units (wall_thickness)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation_item.name='wall thickness of stock'}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
stock_size_parameters to number_-with_units (width)	PATH			measure_representation_item<= measure_with_unit {measure_representation_item<= representation_item representation_item.name='width of stock'}
stock_size_parameters to text (textual_parameter)	PATH			measure_representation_item<= measure_with_unit [measure_with_unit.value_component-> measure_value {measure_value = descriptive_measure}] [measure_with_unit.unit_component-> unit unit=named_unit named_unit=> context_dependent_unit {(context_dependent_unit.name) (context_dependent_unit.name='textual parameter')}]
TDP_ELEMENT #1: When tdp_element is a parts list #2: When tdp_element is a drawing defined within this document #3: When tdp_element is a product_-data_set #4: When tdp_element is a reference_document #5: When tdp_element is a drawing that is externally defined to this document #6: When tdp_element is a drawing and is defined by AP 202	#1,#2,#3,#4,#5: (product_definition_-formation) #6: (<draughting_drawing_revision> <product_definition_formation>)	41 505 41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.of_products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category _relationship.category product_category_relationship product_category_relationship.sub_category-> product_category #1: (product_category.name='parts list') #2: (product_category.name='drawing') #3: (product_category.name='product data set') #4: (product_category.name='reference document') #5: (product_category.name='externally defined document') #6:(<draughting_drawing_revision> <product_category.name='drawing'>)

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT_LIST_ITEM	product_definition	41		
tdp_element_list_item to element_-identification (tdp_element_item)	PATH			product_definition product_definition.formation-> product_definition_formation
tdp_element_list_item to identifier (reference_code)	PATH			product_definition classification_item=product_definition classification_item<- applied_classification_assignment.items[i] applied_classification_assignment<= {classification_assignment role_select=classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='reference group'} classification_assignment classification_assignment.assigned_class-> group group.name {group=> class}

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
TIME_INTERVAL_EFFECTIVITY	time_interval_based_effectivity	41		
time_interval_effectivity to date (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period->time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound->date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)
time_interval_effectivity to date (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period->time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound->date_time_or_event_occurrence date_time_or_event_occurrence=date_time_select (date_time_select=date_and_time) (date_time_select= date)
time_interval_effectivity to event (primary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period->time_interval=> time_interval_with_bounds time_interval_with_bounds.primary_bound->date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
time_interval_effectivity to event (secondary_bound)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.secondary_bound-> date_time_or_event_occurrence date_time_or_event_occurrence=event_occurrence event_ocuurence
time_interval_effectivity to label (interval_name)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval time_interval.name
time_interval_effectivity to number_-with_units (duration)	PATH			time_interval_based_effectivity time_interval_based_effectivity.effectivity_period-> time_interval=> time_interval_with_bounds time_interval_with_bounds.derivation-> measure_with_unit
TRANSFORMATION	axis2_placement	42		

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
VOLUME	measure_with_unit	41		
volume to number_with_units (a_value)	IDENTICAL MAPPING			[measure_with_unit.unit_component-> unit unit=named_unit named_unit=> volume_unit] [named_unit.dimensions-> dimensional_exponents] {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-0} {dimensional_exponents.mass_exponent=0} {dimensional_exponents.length_exponent=3} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}
WEIGHT	representation_item	43		
weight to a_real (a_value)	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value measure_value=numeric_measure

Table 11 — Mapping table for parts_list (continued)

Application element	AIM element	Source	Rules	Reference path
weight to measure (unit_of_measure) #1: If weight is expressed in SI units #2: If weight is not expressed in SI units	PATH			representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.unit_component-> unit unit=named_unit named_unit=> #1: (si_unit si_unit.name-> si_unit_name si_unit_name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]) #2: (conversion_based_unit conversion_based_unit.name [conversion_based_unit.conversion_factor-> measure_with_unit] [measure_with_unit.value_component-> measure_value] [measure_with_unit.unit_component-> unit unit=named_unit named_unit=> si_unit si_unit.name='newton' [named_unit.dimensions-> dimensional_exponents {dimensional_exponents.amount_of_substance_exponent=0} {dimensional_exponents.electric_current_exponent=0} {dimensional_exponents.time_exponent=-2} {dimensional_exponents.mass_exponent=1} {dimensional_exponents.length_exponent=1} {dimensional_exponents.luminous_intensity_exponent=0} {dimensional_exponents.thermodynamic_temperature_exponent=0}]])

Table 11 — Mapping table for parts_list (concluded)

Application element	AIM element	Source	Rules	Reference path
weight to weight_derivation (derivation_method)	PATH			representation_item=> [qualified_representation_item qualified_representation_item.qualifiers[i]-> value_qualifier] [measure_representation_item<= measure_with_unit<- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifier[i]-> value_qualifier] value_qualifier=type_qualifier type_qualifier.name {measure_qualification.name='weight derivation method'} {(type_qualifier.name='calculated') (type_qualifier.name='estimated') (type_qualifier.name='weighed')}

Table 12 — Mapping table for presentation

Application element	AIM element	Source	Rules	Reference path
A_NUMBER	number	11		
A_REAL	real	11		
AN_INTEGER	integer	11		
BODY_PRESENTATION	area_dependent_annotation_-representation	46		
body_presentation to tabulation (body_tabulation_presentation)	IDENTICAL MAPPING			
COLUMNAR_DATA_CONTENT_-HOLDER	area_dependent_annotation_representation	46		
columnar_data_content_holder to field_content_holder_tabulation (column_content_presentation)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
COLUMN_HEADER	area_dependent_annotation_representation	46		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
column_header to field_title (single_column_header_titles)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
DATE	(date)(date_and_time)	41		
date to an_integer (day)	PATH			date=> calendar_date calendar_date.day_component
date to an_integer (month)	PATH			date=> calendar_date calendar_date.month_component
date to an_integer (week)	PATH			date=> week_of_year_and_day_date week_of_year_and_day_date.week component
date to an_integer (year)	date.year_component	41		
date to time (specific_time)	PATH			date<- date_and_time.date_component date_and_time date_and_time.time_component local_time

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
EXPLICIT_GRAPHICS	annotation_occurrence	46		
explicit_graphics to geometry (table_geometry)	PATH			[annotation_occurrence=> annotation_point_occurrence] [annotation_occurrence=> annotation_curve_occurrence]
EXTERNAL_GRAPHICS_FILE	externally_defined_symbol_and_ placement	232		
external_graphics_file to a_real (file_loadpoint_x)	PATH			externally_defined_symbol_and_placement<= placement-> cartesian_point cartesian_point.coordinates[1]
external_graphics_file to a_real (file_loadpoint_y)	PATH			externally_defined_symbol_and_placement<= placement-> cartesian_point cartesian_point.coordinates[2]

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
external_graphics_file to file (file_id)	PATH			(externally_defined_symbol_and_placement<= externally_defined_symbol<= externally_defined_item externally_defined_item.item_id) (externally_defined_symbol_and_placement document_reference_item=externally_defined_symbol_and_placement document_reference_item<- applied_document_reference.items[i] applied_document_reference<= document_reference {document_reference role_select=document_reference role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name=\ 'presentation graphics file'} document_reference.assigned_document-> document=> document_file)
FIELD_CONTENT_HOLDER	annotation_text_occurrence	46		
field_content_holder to label (text_style)	PATH			annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.styles-> presentation_style_assignment presentation_style_assignment.styles[i]-> presentation_style_select presentation_style_select=text_style text_style text_style.name

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
field_content_holder to text (field_justification)	PATH	46	34	annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.alignment {(text_literal.alignment='baseline left') (text_literal.alignment=' baseline centre') (text_literal.alignment='baseline right')}
FIELD_CONTENT_HOLDER_-SINGULAR	annotation_text_occurrence	46		
field_content_holder to text (field_text)	PATH			annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.literal
FIELD_CONTENT_HOLDER_-TABULATION	annotation_text_occurrence	46		
field_content_holder to field_-location (location)	PATH			annotation_text _occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> placement=> axis2_placement_2d

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
field_content_holder to text (field_text)	PATH			annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.literal
FIELD_LOCATION	axis2_placement_2d	42		
FIELD_LOCATION_DEFINITION	axis2_placement_2d	42		
field_location_definition to load_-point (field_load_point)	PATH			axis2_placement_2d<= placement placement.location-> cartesian_point
FIELD_TITLE	annotation_text_occurrence	46		
field_title to field_location (location)	PATH			annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.alignment-> axis2_placement axis2_placement=axis2_placement_2d axis2_placement_2d

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
field_title to label (field_style)	PATH		35	annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.font-> font_select font_select=externally_defined_text_font externally_defined_text_font
field_title to text (field_justification)	PATH		34	annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.alignment {(text_literal.alignment='left') (text_literal.alignment='center') (text_literal.alignment='right')}
field_title to text (field_text)	PATH			annotation_text_occurrence<= annotation_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal text_literal.literal

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
FILE	document_file	232		{document_file<= document<- document_representation_type.representation_types document_representation_type (document_representation_type.name='digital') (document_representation_type.name='physical')}
file to change_identification (change_status) (see note 3)				
file to content_property (file_content_property) (see note 3)				
file to distribution_notice (distribution_authorizations) (see note 3)				
file to file_format (context_file_format)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'}

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
file to identifier (context_file_name) (see note 3)				
file to identifier (native_format_file_name) (see note 3)				
file to label (file_content_type)	PATH			document_file<= document document.kind-> document_type document_type.product_data_type
file to notation (file_note) (see note 3)				
file to person_and_organization (assigned_to) (see note 3)				
file to security_classification (security_identification) (see note 3)				
file to size_characteristics (size) (see note 3)				
file to system_declaration (source_system) (see note 3)				

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_FORMAT	representation	41		{represemtation.name='document format'}
file_format to date (release_date)	PATH			representation (date_item=representation date_item<- applied_date_assignment.items applied_date_assignment<= date_assignment date_assignment.assigned_date->date {date_assignment.role->date_role date_role.name='release date'}) (date_and_time_item=representation date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role->date_time_role date_time_role.name='release date'} date_and_time_assignment.assigned_date_and_time->date_and_time)
file_format to identifier (change_level) (see note 3)				
file_format to identifier (format_code) (see note 3)				

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (revision_level) (see note 3)				
file_format to text (format_standard) (see note 3)				
FILE_RELATIONSHIP	document_relationship	41		
file_relationship to file (related_file)	PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to file (relating_file)	PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to label (name)	PATH			document_relationship document_relationship.name
file_relationship to text (description)	PATH			document_relationship document_relationship.desciption
GEOMETRY	geometric_representation_item	42		
GROUP_PARENT_COLUMN_-HEADER	area_dependent_annotation_representation	46		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
group_parent_column_header to field_title (single_group_parent_column_-header_titles)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
GROUP_PARENT_CONTENT_-HOLDER	area_dependent_annotation_ representation	46		
group_parent_content_holder to field_content_holder_tabulation (group_parent_content_presentation)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
GROUP_PARENT_HEADER	area_dependent_annotation_ representation	46		
group_parent_header to field_title (single_group_parent_header_titles)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
HEADER_BLOCK	area_dependent_annotation_occurrence	46		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
header_block to field_content_-holder_singular (single_block_content_presentation)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
header_block to field_title (single_block_titles)	PATH			annotation_text_occurrence<= styled_item styled_item.item-> representation_item=> geometric_representation_item=> text_literal
HEADER_PRESENTATION	area_dependent_annotation_ representation	46		
header_presentation to header_block (header_blocks_presentation)	IDENTICAL_MAPPING			
header_presentation to tabulation (header_tabulation_presentation)	IDENTICAL_MAPPING			
IDENTIFIER	identifier	41		
LABEL	label	41		
LIST_PRESENTATION	presentation_set	46		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
list_presentation to measure (unit_of_length)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area<= presentation_representation<= representation representation.context_of_items-> representation_context=> global_unit_assigned_context global_unit_assigned_context.units-> unit {unit=named_unit named_unit=> length_unit}
list_presentation to body_ presentation (body_page1_format)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area<= presentation_representation<= representation representation.items-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation=> presentation_representation=> area_dependent_annotation_representation

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
list_presentation to body_ presentation (body_pagen_format)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area<= presentation_representation<= representation representation.items-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation=> presentation_representation=> area_dependent_annotation_representation
list_presentation to header_ presentation (header_page1_format)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area<= presentation_representation<= representation<= representation.items-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation=> presentation_representation=> area_dependent_annotation_representation

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
list_presentation to header_ presentation (header_pagen_format)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area<= presentation_representation<= representation representation.items-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation=> presentation_representation=> area_dependent_annotation_representation
list_presentation to page_ presentation (page_format)	PATH			presentation_set<- area_in_set.in_set area_in_set area_in_set.area-> presentation_area
LOAD_POINT	cartesian_point	42		
load_point to a_real (load_point_x)	PATH			cartesian_point cartesian_point.coordinates[1]
load_point to a_real (load_point_y)	PATH			cartesian_point cartesian_point.coordinates[2]
MEASURE	named_unit	41		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
PAGE_PARAMETERS	presentation_size	46		presentation_size {presentation_size.unit-> presentation_size_assignment_select presentation_size_assignment_select=presentation_area}
page_parameters to a_real (page_anchor_x)	PATH			presentation_size presentation_size.size-> planar_box planar_box.placement-> axis2_placement axis2_placement=axis2_placement_2d axis2_placement_2d<= placement placement.location->cartesian_point cartesian_point.coordinates[1]
page_parameter to a_real (page_anchor_y)	PATH			presentation_size presentation_size.size-> planar_box planar_box.placement-> axis2_placement axis2_placement=axis2_placement_2d axis2_placement_2d<= placement placement.location->cartesian_point cartesian_point.coordinates[2]

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
page_parameters to explicit_graphics (graphics_declaration)	PATH			presentation_size presentation_size.unit-> presentation_size_assignment_select=presentation_area presentation_area<= presentation_representation<= representation representation.items-> representation_item=> styled_item=> annotation_occurrence
page_parameters to external_-graphics_file (graphics_declaration)	PATH			presentation_size presentation_size.unit-> presentation_size_assignment_select= presentation_area<= presentation_representation<= representation representation.items-> representation_item=> geometric_representation_item=> placement=> externally_defined_symbol_and_placement
page_parameter to text (page_size_code)	PATH			presentation_size presentation_size.size-> planar_box=> externally_defined_planar_box<= externally_defined_item externally_defined_item.source-> external_source external_source.source_id {(external_source.source-id) (external_source.source_id='ISO_54-1980') (external_source.source_id='ASME_14/1-80')}

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
page_parameter to text (paper_size)	PATH			presentation_size presentation_size.size-> planar_box=> externally_defined_planar_box<= externally_defined_item externally_defined_item.item_id-> source_item
PAGE_PRESENTATION	presentation_area	46		
page_presentation to page_-parameters (page1_parameters)	PATH			presentation_area presentation_size_assignment_select=presentation_area presentation_size_assignment_select<- presentation_size.unit presentation_size
page_presentation to page_-parameters (page_master_parameters)	PATH			presentation_area presentation_size_assignment_select=presentation_area presentation_size_assignment_select<- presentation_size.unit presentation_size
page_presentation to page_-parameters (pagen_parameters)	PATH			presentation_area presentation_size_assignment_select=presentation_area presentation_size_assignment_select<- presentation_size.unit presentation_size
STRING_LOCATION_-DEFINITION	axis2_placement_2d	42		

Table 12 — Mapping table for presentation (continued)

Application element	AIM element	Source	Rules	Reference path
string_location_definition to load_-point (string_load_point)	PATH			axis2_placement_2d<= placement placement.location->cartesian_point
TABULATION	annotation_text_occurrence	46		
tabulation to column_header (tabulation_columnar_headings)	IDENTICAL MAPPING			
tabulation to columnar_data_-content_holder (tabulation_columns)	IDENTICAL MAPPING			
tabulation to group_parent_column_-header (tabulation_group_parent_headings)	IDENTICAL MAPPING			
tabulation to group_parent_content_-holder (parent_columns)	IDENTICAL MAPPING			
tabulation to group_parent_header (group_parent_headings)	IDENTICAL MAPPING			
tabulation to tabulation_header (tabulation_headings)	IDENTICAL MAPPING			
TABULATION_HEADER	area_dependent_annotation_-representation	46		

Table 12 — Mapping table for presentation (concluded)

Application element	AIM element	Source	Rules	Reference path
tabulation_header to field_title (single_tabulation_header_titles)	PATH			area_dependent_annotation_representation<= presentation_represenation<= representation representation.items-> representation_item[i]=> styled_item=> annotation_occurrence=> annotation_text_occurrence
TEXT	text	41		
TIME	local_time	41		
time to a_real (second)	local_time.second_component	41		
time to an-integer (hour)	local_time.hour_component	41		
time to an_integer (minute)	local_time.minute_component	41		
time to text (zone)	PATH			local_time local_time.zone-> coordinated_universal_time_offset

Table 13 — Mapping table for product_data_set

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_IDENTIFICATION_-OF_ITEM	product_definition_formation	41		
alternate_identification_of_item to change_identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.item[1] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to design_authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name=‘creator’) (person_and_organization_role.name=‘id owner’)}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name=‘creator’) (organization_role.name=‘id owner’)})] #2: <product_definition_formation product_definition_formation.of_product-> product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name=’id owner’)}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
alternate_identification_of_item to drawing_suffix_number_-combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to identifier (identifying_number)	PATH			product_definition_formation [product_definition_formation.id] product_definition_formation.of_product-> product [product.id] <product_defintion_formation oganization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment organization_assignment.assigned_organization-> organization>
alternate_identification_of_item toitem_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category
alternate_identification_of_item to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_identification_of_item to source_information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ANNOTATION	annotation_occurrence	46		
annotation to geometry (annotation_geometry)	PATH			annotation_occurrence<= styled_item {styled_item<= representation_item<- representation.items[i] representation} styled_item.item-> representation_item=> geometric_representation_item

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
CONTENT_PROPERTY	representation	43		{representation<- {[representation.name='document content'] [representation.context_of_items-> representation_context representation_context.context_type='document parameters']} property_definition_representation.used_representation property_definition_representation property_definition_representation.defintion-> represented_definition represented_definition=property_definition property_definition {property_definition.name='document property'}}
content_property to a_real (real_world_scale) #1: real number value #2: text number	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='real world scale'} #1: (measure_representation_item<= measure_with_unit [measure_with_unit.value_component-> measure_value measure_value=ratio_measure] [measure_withunit.unit_component-> unit unit=named_unit named_unit-> ratio_unit]) #2: (descriptive_representation_item descriptive_representation_item.description)
content_property to text (detail_level)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='detail level'} descriptive_representation_item descriptive_representation_item.descrption

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
content_property to text (geometry_type)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='geometry type'} descriptive_representation_item descriptive_representation_item.description
content_property to text (languages) #1: through descrptive text #2: through classification structure	PATH			#1: (representation representation.items[i]-> representation_item=> {representation_item.name='language'} descriptive_representation_item.description) #2: (representation language_item=representation language_item<- language_assignment.items[i] language_assignment<= classification_assignment {classification_assignment.role-> classification_role classification_role.name='language'} classification_assignment.assigned_class-> group {group=> language} group.name)

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
DRAWING #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or AP 202 exchange with and AP 232 exchange	#1: (product_definition_formation #2: ([draughting_drawing_revision] [product_definition_formation])	41 505 41	24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'}) #2: ({draughting_drawing_revision<= drawing_revision<= presentation_set presentation_representation_select=presentation_set presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition product_definition.formation-> product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'})

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
drawing to header (heading) #3: (see note 2)	#1: (PATH) #2: (PATH)			#1,#2: (product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='drawing header'})
drawing to sheet (pages) (see note 3)				
DRAWING_SUFFIX_NUMBER_- COMBINATION	product	41		
drawing_suffix_number_- combination to element_- identification (drawing_number)	PATH	41	13	product<- product_relationship.relating_product product_relationship {product_relationship.name=’drawing suffix number combination’} product_relationship.related_product->product product.id { product<- product_related_product_category.products[i] product_related_product_category<= product_category<- product_categoy.name=’document’ product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category.name=’drawing’}

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_suffix_number_- combination to identifier (suffix_identifier)	PATH		13	product<- product_relationship.relating_product product_relationship product_relationship.id {product_relationship.name='drawing suffix number combination'}
EXTERNAL_LIBRARY_- REFERENCE #1: If external_library_reference is an element in a class source select, that is Reference by a general_-classification as the classification_-source. #2: If external_library_reference is an element in a property_source _-select, that is reference by a property as the property_source.	#1: (externally_defined_class) #2: (externally_defined_general_-property)	232 232		#1: (externally_defined_class<= [ckass<= group][externally_defined_item]) #2:(externally_defined_general_property<= [general_property] [externally_defined_item])
external_library_reference to identifier (external_id)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.item_id
external_library_reference to identifier (library_type)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source-> external_sourrce external_sourrce.id {external_source.source_id-> source_item source_item=identifier identifier}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
external_library_reference to text (description)	PATH			#1: (externally_defined_class<=) #2: (externally_defined_general_property<=) externally_defined_item externally_defined_item.source->exterrnal_source external_sourcee.description
FILE	document_file	232		{document_file<= document<- document_representation_type.representation_types document_representation_type (document_representation_type.name='digital') (document_representation_type.name='physical')}
file to change_identification (change_status)	PATH			document_file action_item=document_file action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='change identification'} action_assignment.assigned_action->action=> executed_action

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file to distribution_notice (distribution_authorizations)	PATH			document_file approval_item=document_file approval_item<- applied_approval_assignment.items[i] applied_approval_assignment<= approval_assignment {approval_assignment<- role_select=approval_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='distribution notice'} approval_assignment.assigned_approval->approval
file to file_format (context_file_format)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='document format'}
file to identifier (context_file_name)	PATH			document_file<= document document.id

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file to identifier (native_format_file_name)	PATH			(document_file<= document<- document_relationship.related_document document_relationship {document_relationship.name='translation relationship'} document_relationship.relating_document-> document {document=> document_file} document.id
file to label (file_content_type)	PATH			document_file<= document document.kind-> document_type document_type.product_data_type
file to notation (file_note)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='notation'} representation.items[i]-> representation_item=> descriptive_representation_item

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file to security_classification (security_identifications)	PATH			document_file security_classification_item=document_file security_classification_item<- applied_security_classification_assignment.items[i] applied_security_classification_assignment<= security_classification_assignment <security_classification_assignment role_select=security_classification_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role> security_classification_assignment.assigned_security_classification-> security_classification
file to size_characteristics (size)	PATH			document_file<= characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file to system_declaration (source_system)	PATH			document_file characterized_object characterized_definition=characterized_object characterized_definition<- property_definition.definition property_definition {property_definition.name='document property'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {(representation.name='source system declaration') (representation.name='document creation')}

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_FORMAT	representation	41		{representation.name='document format'}
file_format to date (release_date)	PATH			representation (date_item=representation date_item<- applied_date_assignment.items applied_date_assignment<= date_assignment date_assignment.assigned_date->date {date_assignment.role->date_role date_role.name='release date'}) (date_and_time_item=representation date_and_time_item<- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment<= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role->date_time_role date_time_role.name='release date'} date_and_time_assignment.assigned_date_and_time->date_and_time)

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (change_level)	PATH			representation action_item=representation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action {action=> executed_action} action.name {action.chosen_method-> action_method action_method.name='change level'}
file_format to identifier (format_code)	PATH			representation representation.items[i]-> representation_item=> {representation_item.name='character code'} descriptive_representation_item descriptive_representation_item.description

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
file_format to identifier (revision_level)	PATH			representation action_item=representation acton_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action action.name {action=> executed_action) {action.chosen_method-> action_method action_method.name='revision level'}
file_format to text (format_standard)	PATH			representation {representation.name='file format'} representation.items[i]-> representation_item=> {representation_item.name='data format'} descriptive_representation_item descriptive_representation_item.description

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
FILE_RELATIONSHIP	document_relationship	41		
file_relationship to file (related_file)	:PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to file (relating_file)	PATH			document_relationship document_relationship.related_document->document=> document_file
file_relationship to label (name)	PATH			document_relationship document_relationship.name
file_relationship to text (description)	PATH			document_relationship document_relationship.description
GEOMETRIC_VALIDATION_-PROPERY	representation	43		

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_validation_property to a_-real (property_value)	PATH			representation representation.items[i]->rpresentation_item {(representation_item.name) ([representation_item.name='surface area'] [representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value=area_measure area_measure[] ([representation_item.name='volume'] [representation_item=> measure_representation_item<= measure_with_unit measure_with_unit.value_component-> measure_value=volume_measure volume_measure]) ([representation_item.name='centroid'] [representation_item=> geometric_representation_item=> point=> cartesian_point])}
geometric_validation_property to label (property_type)	representation.name	43		{(representation) (representation_name='surface area') (representation.name='volume') (representation.name='centroid')}
GEOMETRY	geometric_representation_item	42		
INDEPENDENT_PROPERTY	general_property	41		
independent_property to external_-library_reference (property_source)	IDENTICAL MAPPING			{general_property=> externally_defined_general_property}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property to identifier (id)	general_property.id	41		
independent_property to measure (allowed_unit) #1: If the unit has a commonly used descriptor. #2: If the unit descriptor is constructed of more than one unit name.	PATH			general_property represented_definition=general_property represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation->representation {representation.name='allowed units'} representation.context_of_item->representation_context=> global_unit_assigned_context.units[i]->unit #1: (unit=named_unit named_unit) #2: (unit=derived_unit derived_unit)
independent_property to text (description)	general_property.description	41		
INDEPENDENT_PROPERTY_-RELATIONSHIP	general_property_relationship	41		
independent_property_relationship to text (description)	general_property_-relationship.description	41		
independent_property_relationship to independent_property (related_property)	PATH			general_property_relationship general_property_relationship.related_property->general_property

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
independent_property_relationship to independent_property (relating_property)	PATH			general_property_relationship general_property_relationship.relating_property->general_property
independent_property_relationship to label (relation_type)	general_property_relationship.name	41		
INDEPENDENT_PROPERTY_-USAGE	general_property_association	41		
independent_property_usage to independent_property (assigned_-property)	PATH			general_property_association general_property_association.base_definition->general_property
independent_property_usage to property_assignment (assigned_to)	PATH			general_property_association general_property_association.derived_definition->derived_property_select derived_property_select=property_definition property_definition

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification->certification

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to change_- identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role->object_role object_role.name='change identification'} action_assignment.assigned_action->action=> {action.description='change identification'} executed_action

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.producst[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
ITEM_TYPE	product_category	41		
item_type to company (code_administrator)	PATH			product_category organiation_item=product_category organizatin_item<- assigned_organization_assignment.item[i] assigned_organization_assignment=> organization_assignment {organization_assignment organization_assignment.role-> organization_role organization_role.name='code administator'} organization_assignment.assigned_organization-> organization
item_type to label (item_code)	PATH			product_category product_category.name
item_type to text (type_of_coding_scheme)	PATH			product_category product_category.description

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set'}

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to drawing (related) #1: When drawing is identified within AP 232 #2: When drawing is defined as an AP 201 or AP 202 exchange within the same file	PATH			#1:(product_definition_formation<- {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related drawing'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'}) #2: (product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product-category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set'}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related drawing'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='drawing'} presented_item_item=product_definition_formation presented_item_item<- applied_presented_item.items[i] applied_presented_item<= presented_item<- presented_item_representation.item presented_item_representation presented_item_representation.presentation-> presentation_representation_select presentation_representation_select=presentation_set presentation_set=> drawing_revision=> draughting_drawing_revision)

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to product_data_set (related)	PATH			product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set'} product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {product_definition_formation_relationship.name='related product data set'} product_definition_formation_relationship.related_product_definition_formation -> product_definition_formation {product_definition_formation.of_product-> product<- product_related_product_category.products product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category {product_category.name='product data set'}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set to source_file (file_configuration)	PATH			product_definition_formation<- product_definition.formation (product_definition=> product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i]-> document=> document_file) (product_defintion document_reference_item=product_defintion document_reference_item<- applied_document_reference.item[i] applied_document_reference<= document_reference=> document_file)
PRODUCT_DATA_SET_WITH_-FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set with format'}
product_data_set_with_format to header (product_header)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='product data set with format header'}

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_with_format to product_presentation (presentation_of_product_data_set)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation=> (presentation_representation=> mechanical_design_geometric_presentation_representation) (presentation_representation)
PRODUCT_DATA_SET_WITH_-SHADING	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set with shading'}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_with_shading to header_configuration_with_-element_identification (data_configuration)	PATH			product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description=\ 'product data set with shading header'}
product_data_set_with_shading to shaded_shape_model (a_shaded_model)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation=> (presentation_representation=> mechanical_design_shaded_presentation_representation) (presentation_representation)

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_DATA_SET_- WITHOUT_FORMAT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category='product data set without format'}
product_data_set_without_format to header_configuration with_- element_identification (data_configuration)	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name=\ 'product data set without format header'}

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_data_set_without_format to shape_model (a_model) #1: Allows ability to apply global uncertainty value to shape model	PATH			product_defnition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation=> shape_representation #1: <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit>

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_PRESENTATION #1: For general capability #2: For use with AIC 517	#1: (presentatation_representation) #2: (mechanical_design_geometric_- presentation_representation)	46 517		
product_presentation to annotation (annotation_presentation) #2: (see note 2)	#1: (PATH)			#1: (presentation_representation<= representation representation.items[i]->) representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation representation.items[i]-> styled_item=> annotation_occurrence)

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
product_presentation to shape_model (part_shape_views)	PATH			#1: (presentation_representation<= representation representation.items[i]->) representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_presentation-> representation <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit> (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation) (shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation) (shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation))

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation=> shape_representation) (representation representation.items[i]-> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]-> advanced_face)) #2: (mechanical_design_geometric_presentation_representation<= presentation_representation<= representation representation.items[i]-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_representation-> representation <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit> (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation)

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation) (shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation)) (representation=> shape_representation) (representation representation.items[i]-> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]-> advanced_face))
PROPERTY_ASSIGNMENT	property_definition	41		
property_assignment to identifiier (described_element)	PATH			property_definition property_definition.definition-> characterized_definition
property_assignment to label (name)	property_definition.name	41		
property_assignment to text (description)	property_definition.description	41		

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
SHADED_SHAPE_MODEL	#1: (presentation_representation) #2: (mechanical_design_shaded_- presentation_represenation)	46 518		

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
shaded_shape_model to shape_-model (part_shape_views) #3: Allows ability to apply global uncertainty value to shape model.	PATH			#1: (presentation_representation<= representation representation.items[i]->representation_item=> mapped_item mapped_item.mapping_source->representation_map representation_map.mapped_presentation->representation #3: <representation=> shape_representation. shape_representation.context_of_items[i]->representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]->uncertainty_measure_with_unit> (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation) (shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation) (shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation))

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation=> shape_representation) (representation representation.items[i]-> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]-> advanced_face)) #2: (mechanical_design_shaded_presentation_representation<= presentation_representation<= representation representation.items[i]-> representation_item=> mapped_item mapped_item.mapping_source-> representation_map representation_map.mapped_presentation-> representation <representation=> shape_representation. shape_representation.context_of_items[i]-> representation_context global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i]-> uncertainty_measure_with_unit> (representation=> shape_representation (shape_representation=> edge_based_wireframe_shape_representation) (shape_representation=> shell_based_wireframe_shape_representation) (shape_representation=> geometrically_bounded_2d_wireframe_representation) (shape_representation=> geometrically_bounded_surface_shape_representation) (shape_representation=> non-manifold_surface_shape_representation) (shape_representation=> manifold_surface_shape_representation)

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_representation=> geometrically_bounded_wireframe_surface_shape_representation) (shape_representation=> faceted_brep_shape_representation) (shape_representation=> elementary_brep_shape_representation) (shape_representation=> advanced_brep_shape_representation) (shape_representation=> csg_shape_representation)) (representation=> shape_representation) (representation representation.items[i]-> representation_item=> topological_representation_item=> connected_face_set connected_face_set.cfs_faces[i]-> advanced_face))

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
SHAPE_MODEL	#1: (edge_based_wireframe_shape_-representation)	501		
	#2: (shell_based_wireframe_shape_-representation)	502		
	#3: (geometrically_bounded_2d_-wireframe_representation)	503		
	#4: (geometrically_bounded_surface_-shape_representation)	507		
	#5: (non-manifold_surface_shape_-representation)	508		
	#6: (manifold_surface_shape_-representation)	509		
	#7: (geometrically_bounded_wireframe_-surface_shape_representation)	510		
	#8: (advanced_face)	511		
	#9: (faceted_brep_shape_representation)	512		
	#10: (elementary_brep_shape_-representation)	513		
	#11: (advanced_brep_shape_-representation)	514		
	#12: (csg_shape_representation)	515		
	#13: (shape_representation)	41		

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
shape_model to geometric_- validation_property (validation_properties)	PATH			representation<- {representation=> shape_representation} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition-> represented_definition represented_definition=property_definition property_definition {property_definition.name='geometric validation property'} <property_definition.definition-> characterzed_definition characterized_definition=shape_definition shape_definition #1: (shape_definition=product_definition_shape product_definition_shape) #2: (shape_definition=shape_aspect shape_aspect)> property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
shape_model to geometry (part_shape_representation)	PATH			#1: (edge_based_wireframe_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #2: (shell_based_wireframe_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #3: (geometrically_bounded_2d_wireframe_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #4: (geometrically_bounded_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #5: (non-manifold_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #6: (manifold_surface_shape_representation<= shape_represenation<= representation representation.items[i]-> representation_item=> geometric_representation_item) #7: (geometrically_bounded_wireframe_surface_shape_representation<= shape_representation<= representation representation.items[i]->

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item=> geometric_representation_item) #8: (advanced_face<- connected_face_set.cfs_faces[i]->) connected_face_set<= topological_representation_item<= representation_item=> geometric_representation_item) #9: (faceted_brep_shape_representation<= shape_represenation<= representation representation.items[i]->) representation_item=> geometric_representation_item) #10: (elementary_brep_shape_representation<= shape_represenation<= representation representation.items[i]->) representation_item=> geometric_representation_item) #11: (advanced_brep_shape_representation<= shape_represenation<= representation representation.items[i]->) representation_item=> geometric_representation_item) #12: (csg_shape_representation<= shape_represenation<= representation representation.items[i]->) representation_item=> geometric_representation_item) #13:(shape_representation<= representation representation.items[i]->) representation_item=> geometric_representation_item)

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
SHEET #1: If sheet is identified within AP 232 #2: If sheet is defined as an AP 202 exchange within same file	#1: product_definition_formation #2: [product_definition_formation] [drawing_sheet_revision]	41 41 101	15 18 24	#1: ({product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'}) #2: ({drawing_sheet_revision<= presentation_area<= presentation_representation presentation_representation_select=presentation_representation presentation_representation_select<- presented_item_representation.presentation presented_item_representation presented_item_representation.item-> presented_item=> applied_presented_item applied_presented_item.items[i]-> presented_item_item presented_item_item=product_definition_formation product_definition_formation} {product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category {product_category.name='document'} product_category<- product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='sheet'})

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to configuration (sheet_configuration)	PATH			#1, #2 : product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'})
sheet to element_identification (sheet_identifier)	IDENTICAL MAPPING			
sheet to label (sheet_size)	PATH			#1, #2: product_definition_formation<- product_definition.formation product_definition {product_definition.description='sheet content'} characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.description='sheet header'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation {representation.name='document format'} representation.items[i]-> {(representation_item.name='size format') (representation_item.name='size format standard')} representation_item=> descriptive_representation_item descriptive_representation_item.description

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
sheet to product_presentation (presentation_format) (see note 3)				
SIZE_CHARACTERISTICS_-INTERNAL_DIVISIONS	representation	43		{representation.name=’multiple sheets in one file’)
size_characteristics_internal_division to an_integer (internal_division_count)	PATH			representation representation.items[i]->representation_item {representation_item.name='internal division count'} representation_item=>measure_representation_item
size_characteristics_internal_division to label (internal_division_type)	PATH			representation representation.items[i]->representation_item {representation_item.name='internal division type'} representation_item {representation_item.name='sheet size'} representation_item=>descriptive_representation_item.description
SIZE_CHARACTERISTICS_-SHEET_ACROSS_FILE	representation	43		{representation.name=’file containing portion of sheet’}
size_characteristics_sheet_across_-file to an_integer (frame_number_-comprising_this_file)	PATH			representation representation.items[i]->representation_item {representation_item.name='sheet portion number'} representation_item=>measure_representation_item

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
size_characteristics_sheet_across_-file to an_integer (number_of_frames_for_sheet)	PATH			representation representation.items[i]-> representation_item {representation_item.name=\ 'number of portions sheet is divided into'} representation_item=> measure_representation_item
SOURCE_FILE	document_file	232		
SPECIAL_CONDITION	descriptive_representation_item	45		
special_condition to label (code)	PATH			descriptive_representation_item<= representation_item representation_item.name
special_condition to text (description)	PATH			descriptive_representation_item descriptive_representation_item.description
special_condition to text (type_of_coding_scheme)	PATH			descriptive_representation_item<= representation_item<- representation.items[i] representation {representation.name='special condition'} representation.context_of_items[i]-> representation_context representation_context.context_type
SYSTEM_DECLARATION	representation			{(representation.name='source system declaration') (representation.name='destination system declaration') (representation.name='document creation')}

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to identifier (system) #1: Use when system operating system, creating_interface, and system_related_element_identifier are needed. #2 : Use when system operating system, creating_interface, and system_related_element_identifier are not needed.	PATH			#1: (representation representation.items[i]->representation_item=> {(representation_item.name='computer system') (representation_item.name='creating system')} descriptive_representation_item descriptive_representation_item.description) #2: (document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment [external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='source system') (identification_role.description='destination system')}] [external_identification_assignment.source->external_source external_source.source_id])

Table 14 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to identifier (system_element_identifier) #1: Use when system operating system, creating_interface, and system_related_element_identifier are needed. #2 : Use when system operating system, creating_interface, and system_related_element_identifier are not needed.	PATH			#1: (representation representation.items[i]->representation_item=> {representation_item.name='element on computer system'} descriptive_representation_item descriptive_representation_item.description) #2: (document_file external_identification_item=document_file external_identification_item<- applied_external_identification_assignment.items[i] applied_external_identification_assignment<= external_identification_assignment<= identification_assignment {identification_assignment.role->(identification_role) (identification_role.description='access context')} identification_assignment.assigned_id)
system_declaration to identifier (system_related_element_identifier)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='related element on computer system'} descriptive_representation_item descriptive_representation_item.description
system_declaration to text (creating_interface)	PATH			representation representation.items[i]->representation_item=> {representation_item.name='creating interface'} descriptive_representation_item descriptive_representation_item.description

Table 13 — Mapping table for product_data_set (continued)

Application element	AIM element	Source	Rules	Reference path
system_declaration to text (operating system)	PATH			representation representation.items[i]->representation_item=>{representation_item.name='operating system'} descriptive_representation_item descriptive_representation_item.description

Table 13 — Mapping table for product_data_set (concluded)

Application element	AIM element	Source	Rules	Reference path
TDP_ELEMENT #1: When tdp_element is a product_-data_set #2: When tdp_element is a reference_document #3: When tdp_element is a drawing	#1,#2: product_definition_formation	41	24	#1:({product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category<- product_category.name='product data set' }) #2:({product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category<- product_category.name='reference document'}) #3:({product_definition_formation product_definition_formation.of_product->product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category->product_category<- product_category.name='drawing'})

Table 14 — Mapping table for reference_document

Application element	AIM element	Source	Rules	Reference path
ITEM_IDENTIFICATION	product_definition_formation	41		
item_identification to alternate_-identification_of_item (alternate_identifications) #1: Use for version to version alternate identification such as supplied parts. #2: Use for part non version alternate identification. #3: When alternate identification is for a combination of version and id #4: When alternate identificationn is for supplier id and version of document.	PATH			#1:(product_definition_formaton<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship {(product_definition_formation_relationship.name) #3(product_definition_formation_relationship.name='alternate part identification') #4: (product_definition_formation_relationship.name='supplied item')} product_definition_formation_relationship.related_product_definition_formation->product_definition_formation) #2:(product_definition_formation->product<- product_relationship.relating_product product_relationship {product_relationship.name='alternate part identification'} product_relationship.related_product->product<- product_definition_formation)
item_identification to certification (item_certification)	PATH			product_definition_formation<- product_definition_formation_relationship.relating_product_definition_formation product_definition_formation_relationship certification_item=product_definition_formation_relationship certification_item<- applied_certification_assignment.items[i] applied_certification_assignment<= certification_assignment certification_assignment.assigned_certification->certification

Table 14 — Mapping table for reference_document (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to change_-identification (change_status)	PATH			product_definition_formation action_item=product_definition_formation action_item<- applied_action_assignment.items[i] applied_action_assignment<= action_assignment {action_assignment role_select=action_assignment role_select<- role_association.item_with_role role_association role_association.role-> object_role object_role.name='change identification'} action_assignment.assigned_action-> action=> {action.description='change identification'} executed_action

Table 14 — Mapping table for reference_document (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to design_- authority (design_activity) #1: used to identify creator of version #2: used to identify owner of document #3: used to identify creator of document representation view. #4: person and organization #5: just organization	PATH		25	#1: [product_definition_formation #4: (person_and_organization_item=product_definition_formation person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator') (person_and_organization_role.name='id owner')}) #5: (organization_item=product_definition_formation organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role (organization_role.name) (organization_role.name='creator') (organization_role.name='id owner')})] #2: <product_definition_formation product_definition_formation.of_product->product #4: (person_and_organization_item=product person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role->person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='id owner')}) #5: (organization_item=product organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role->organization_role

Table 14 — Mapping table for reference_document (continued)

Application element	AIM element	Source	Rules	Reference path
				(organization_role.name) (organization_role.name='id owner')})> #3: <product_definition_formation<- product_definition.formation product_definition #4: (person_and_organization_item=product_definition person_and_organization_item<- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment<= person_and_organization_assignment {person_and_organization_assignment.role-> person_and_organization_role (person_and_organization_role.name) (person_and_organization_role.name='creator')}) #5: (organization_item=product_definition organization_item<- applied_organization_assignment.items[i] applied_organization_assignment<= organization_assignment {organization_assignment.role-> organization_role (organization_role.name) (organization_role.name='creator')})>
item_identification to drawing_- suffix_number_combination (identifying_number)	PATH			product_defintion_formation product_definition_formation.of_product-> product

Table 14 — Mapping table for reference_document (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to identifier (alternate_identifications) #1: alias for version of item #2: alias for id of item #3: alias for product view of item	PATH			#1:(product_definition_formation identifcation_item=product_definition_formation) #2: (product_definition_formation product_definition_formation.of_product-> product identification_item=product) #3: (product_definition_formation<- product_definition.formation product_definition identification_item=product_definition) identification_item<- applied_identification_assignment.item[i] applied_identification_assignment<= identification_assignment identification_assignment.assigned_id {identification_assignment.role-> identification_role identification_role.name='alias'}
item_identification to identifier (identifying_number)	PATH			product_definition_formation product_definition_formation.of_product-> product product.id
item_identification to item_type (classifications)	PATH			product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.producst[i] product_related_product_category<= product_category
item_identification to label (nomenclature_or_name)	PATH			product_definition_formation product_definition_formation.of_product-> product product.name

Table 14 — Mapping table for reference_document (continued)

Application element	AIM element	Source	Rules	Reference path
item_identification to source_- information_type (source_information)	PATH		22	product_definition_formation<- product_definition.formation product_definition characterized_product_definition=product_definition characterized_definition=characterized_product_definition characterized_definition<- property_definition.definition property_definition {property_definition.name='source information'} represented_definition=property_definition represented_definition<- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation-> representation
SPECIFICATION_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='specification document'}

Table 14 — Mapping table for reference_document (concluded)

Application element	AIM element	Source	Rules	Reference path
STANDARD_DOCUMENT	product_definition_formation	41	24	{product_definition_formation product_definition_formation.of_product-> product<- product_related_product_category.products[i] product_related_product_category<= product_category<- {product_category.name='document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category<- {product_category.name='reference document'} product_category_relationship.category product_category_relationship product_category_relationship.sub_category-> product_category product_category.name='standard document'}
TDP_ELEMENT	product_definition_formation	41		

NOTE 2 Application object attribute is not valid for the numbered alternative.

NOTE 3 Application object attribute is not required in the context of the Unit of Functionality.

Rules in the Mapping Table

1) access_file_requires_external_identification
2) approval_requires_approval_date_time
3) approval_requires_approval_person_organization
4) change_identification_restricts_executed_action
5) contract_submission_requires_date_or_date_time
6) contract_submission_requires_destination_organization
7) data_definition_entry_string_restrict_for_superseded_element
8) data_definition_exchange_categorized_as_document
9) data_definition_exchange_restrict_to_one_presentation
10) dependent_instantiable_named_unit
11) distribution_notice_approval_requires_supporting_data
12) document_product_equivalence_existence_rule
13) drawing_suffix_number_combination_identification_constraint
14) exchange_reason_existence_required
15) existence_dependent_drawing_sheet_revision
16) file_format_restricted_to_one_data_definition_entry
17) header_configuration_restricts_property_definition
18) identification_of_sheet_constraint
19) indentured_data_list_identification_constraint
20) indentured_level_tag_identification_constraint
21) indentured_list_method_identification_constraint
22) item_source_information_identification_constraint
23) notation_type_identification_constraint
24) product_requires_category
25) product_version_requires_person_organization
26) reference_document_requires_subcategorization
27) release_authentication_string_restriction_approval_role
28) release_authentication_string_restriction_company_code
29) release_authentication_string_restriction_date_time_role
30) security_classification_date_string_restriction
31) security_declassification_date_string_restriction
32) simple_list_of_elements_constraint
33) simple_list_of_files_constraint
34) text_literal_alignment_baseline_constraint
35) text_literal_font_requires_externally_defined_text_font

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and the AICs and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the text for constructs that are imported from the integrated resources and the AICs. The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes that are not imported into the AIM. Requirements stated in the integrated resources that refer to select list items and subtypes apply exclusively to those items that are imported into the AIM.

The following EXPRESS declaration begins the technical_data_packaging and identifies necessary external references.

EXPRESS specification:

```
*)
SCHEMA technical_data_packaging;

USE FROM action_schema                                  -- ISO 10303-41
  (action,
   action_directive,
   action_method,
   action_relationship,
   action_request_solution,
   action_request_status,
   action_resource,
   action_resource_relationship,
   action_resource_type,
   action_status,
   directed_action,
   executed_action,
   versioned_action_request);

USE FROM aic_advanced_brep;                             -- ISO 10303-514

USE FROM aic_csg;                                       -- ISO 10303-515

USE FROM aic_drawing_structure_and_administration       -- ISO 10303-505
  (draughting_approval_assignment,
   draughting_drawing_revision);

USE FROM aic_edge_based_wireframe;                      -- ISO 10303-501

USE FROM aic_elementary_brep;                           -- ISO 10303-513

USE FROM aic_faceted_brep;                              -- ISO 10303-512

USE FROM aic_geometrically_bounded_2d_wireframe;        -- ISO 10303-503

USE FROM aic_geometrically_bounded_surface;             -- ISO 10303-507

USE FROM aic_geometrically_bounded_wireframe;           -- ISO 10303-510

USE FROM aic_manifold_surface;                          -- ISO 10303-509

USE FROM aic_mechanical_design_geometric_presentation;-- ISO 10303-517
```

```
USE FROM aic_mechanical_design_shaded_presentation;   -- ISO 10303-518

USE FROM aic_non_manifold_surface;                    -- ISO 10303-508

USE FROM aic_shell_based_wireframe;                   -- ISO 10303-502

USE FROM aic_topologically_bounded_surface;           -- ISO 10303-511

USE FROM application_context_schema                   -- ISO 10303-41
  (application_context,
   application_context_relationship,
   application_protocol_definition,
   library_context,
   product_concept_context,
   product_context,
   product_definition_context);

USE FROM approval_schema                              -- ISO 10303-41
  (approval,
   approval_date_time,
   approval_person_organization,
   approval_relationship,
   approval_status);

USE FROM basic_attribute_schema                       -- ISO 10303-41
  (description_attribute,
   id_attribute,
   name_attribute,
   role_association);

USE FROM configuration_management_schema              -- ISO 10303-44
  (configuration_design,
   configuration_effectivity,
   configuration_item);

USE FROM certification_schema                         -- ISO 10303-41
  (certification,
   certification_type);

USE FROM contract_schema                              -- ISO 10303-41
  (contract,
   contract_relationship);

USE FROM date_time_schema                             -- ISO 10303-41
  (calendar_date,
   date_and_time,
   date_time_select,
   event_occurrence,
   event_occurrence_relationship,
   relative_event_occurrence,
   time_interval,
   time_interval_relationship,
   time_interval_with_bounds);

REFERENCE FROM date_time_schema                       -- ISO 10303-41
  (leap_year);

USE FROM document_schema                              -- ISO 10303-41
  (document,
```



```
   document_product_association,
   document_relationship,
   document_representation_type,
   document_type,
   document_usage_constraint,
   product_or_formation_or_definition);

USE FROM draughting_element_schema                    -- ISO 10303-101
  (draughting_callout);

USE FROM drawing_definition_schema                    -- ISO 10303-101
  (draughting_title,
   drawing_revision,
   drawing_sheet_revision);

USE FROM effectivity_schema                           -- ISO 10303-41
  (dated_effectivity,
   effectivity_relationship,
   lot_effectivity,
   serial_numbered_effectivity,
   time_interval_based_effectivity);

USE FROM external_reference_schema                    -- ISO 10303-41
  (externally_defined_item,
   message);

REFERENCE FROM geometry_schema                        -- ISO 10303-42
  (dummy_gri);

USE FROM geometry_schema                              -- ISO 10303-42
  (cartesian_transformation_operator_2d,
   geometric_representation_item
   oriented_surface,
   placement,
   surface_curve);

USE FROM group_schema                                 -- ISO 10303-41
  (group,
   group_relationship);

USE FROM management_resources_schema                  -- ISO 10303-41
  (action_assignment,
   action_request_assignment,
   approval_assignment,
   certification_assignment,
   classification_assignment,
   contract_assignment,
   date_and_time_assignment,
   date_assignment,
   document_reference,
   document_usage_constraint_assignment,
   effectivity_assignment,
   effectivity_context_assignment,
   event_occurrence_assignment,
   external_identification_assignment,
   identification_assignment,
   identification_assignment_relationship,
   organization_assignment,
   organizational_project_assignment,
```

```
   person_and_organization_assignment,
   security_classification_assignment);

USE FROM material_property_definition_schema          -- ISO 10303-45
  (material_designation,
   property_definition_relationship);

USE FROM measure_schema                               -- ISO 10303-41
  (amount_of_substance_measure,
   amount_of_substance_measure_with_unit,
   amount_of_substance_unit,
   area_measure,
   area_measure_with_unit,
   area_unit,
   celsius_temperature_measure,
   context_dependent_measure,
   context_dependent_unit,
   conversion_based_unit,
   count_measure,
   derived_unit,
   descriptive_measure,
   electric_current_measure,
   electric_current_measure_with_unit,
   electric_current_unit,
   global_unit_assigned_context,
   length_measure,
   length_measure_with_unit,
   length_unit,
   luminous_intensity_measure,
   luminous_intensity_measure_with_unit,
   luminous_intensity_unit,
   mass_measure,
   mass_measure_with_unit,
   mass_unit,
   measure_value,
   measure_with_unit,
   named_unit,
   numeric_measure,
   parameter_value,
   plane_angle_measure,
   plane_angle_measure_with_unit,
   plane_angle_unit,
   positive_length_measure,
   positive_plane_angle_measure,
   positive_ratio_measure,
   ratio_measure,
   ratio_measure_with_unit,
   ratio_unit,
   si_unit,
   solid_angle_measure,
   solid_angle_measure_with_unit,
   solid_angle_unit,
   thermodynamic_temperature_measure,
   thermodynamic_temperature_measure_with_unit,
   thermodynamic_temperature_unit,
   time_measure,
   time_measure_with_unit,
   time_unit,
   volume_measure,
```



```
   volume_measure_with_unit,
   volume_unit);

USE FROM method_definition_schema                    -- ISO 10303-49
   (action_method_with_associated_documents);

USE FROM person_organization_schema                  -- ISO 10303-41
   (address,
   organization,
   organization_relationship,
   organizational_address,
   organizational_project,
   organizational_project_relationship,
   person,
   person_and_organization,
   person_organization_select,
   personal_address);

USE FROM presentation_appearance_schema              -- ISO 10303-46
   (character_glyph_style_outline,
   character_glyph_style_stroke,
   externally_defined_style,
   styled_item,
   text_style,
   text_style_for_defined_font);

USE FROM presentation_definition_schema              -- ISO 10303-46
  (annotation_occurrence,
   annotation_curve_occurrence,
   annotation_point_occurrence,
   annotation_text,
   annotation_text_character,
   annotation_text_occurrence,
   composite_text,
   defined_character_glyph,
   externally_defined_character_glyph,
   externally_defined_symbol,
   pre_defined_character_glyph,
   text_literal,
   text_string_representation);

USE FROM presentation_organization_schema            -- ISO 10303-46
  (area_dependent_annotation_representation,
   camera_model,
   camera_usage,
   presentation_area,
   presentation_representation,
   presentation_representation_select,
   presentation_set,
   presented_item,
   presented_item_representation);

USE FROM presentation_resource_schema                -- ISO 10303-46
  (character_glyph_symbol,
   externally_defined_text_font,
   pre_defined_text_font);

USE FROM process_property_schema                     -- ISO 10303-49
  (action_property,
```

```
   product_definition_process,
   process_product_association);

USE FROM process_property_representation_schema       -- ISO 10303-49
  (action_property_representation);

USE FROM product_concept_schema                       -- ISO 10303-44
  (product_concept);

USE FROM product_definition_schema                    -- ISO 10303-41
  (product,
   product_category,
   product_category_relationship,
   product_definition,
   product_definition_context_association,
   product_definition_effectivity,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_formation_with_specified_source,
   product_definition_relationship,
   product_definition_substitute,
   product_definition_with_associated_documents,
   product_related_product_category,
   product_relationship);

USE FROM product_property_definition_schema           -- ISO 10303-41
  (characterized_object,
   characterized_product_definition,
   general_property,
   general_property_association,
   general_property_relationship,
   product_definition_shape,
   property_definition,
   shape_aspect,
   shape_aspect_relationship,
   shape_definition);

USE FROM product_property_representation_schema       -- ISO 10303-41
  (context_dependent_shape_representation,
   item_identified_representation_usage,
   property_definition_representation,
   shape_definition_representation,
   shape_representation,
   shape_representation_relationship);

USE FROM product_structure_schema                     -- ISO 10303-44
  (alternate_product_relationship,
   assembly_component_usage,
   assembly_component_usage_substitute,
   assembly_component_usage_substitute_with_ranking,
   make_from_usage_option,
   next_assembly_usage_occurrence,
   product_definition_occurrence_relationship,
   promissory_usage_occurrence,
   quantified_assembly_component_usage,
   specified_higher_usage_occurrence);

USE FROM qualified_measure_schema                     -- ISO 10303-45
  (descriptive_representation_item,
```



```
   measure_qualification,
   measure_representation_item,
   precision_qualifier,
   qualified_representation_item,
   type_qualifier,
   value_qualifier);

USE FROM representation_schema                          -- ISO 10303-43
  (compound_representation_item,
   compound_item_definition,
   global_uncertainty_assigned_context,
   item_defined_transformation,
   list_representation_item,
   mapped_item,
   representation,
   representation_context,
   representation_item,
   representation_item_relationship,
   representation_map,
   representation_relationship,
   representation_relationship_with_transformation,
   uncertainty_measure_with_unit,
   value_representation_item);

USE FROM security_classification_schema             -- ISO 10303-41
  (security_classification,
   security_classification_level);

REFERENCE FROM support_resource_schema              -- ISO 10303-41
   (label);

REFERENCE FROM topology_schema                          -- ISO 10303-42
  (dummy_tri);
(*

   NOTE - The schemas referenced above can be found in the following parts
   of ISO 10303:

   action_schema                                  ISO 10303-41
   aic_advanced_brep                              ISO 10303-514
   aic_csg                                        ISO 10303-515
   aic_drawing_structure_and_administration       ISO 10303-505
   aic_edge_based_wireframe                       ISO 10303-501
   aic_elementary_brep                            ISO 10303-513
   aic_faceted_brep                               ISO 10303-512
   aic_geometrically_bounded_2d_wireframe         ISO 10303-503
   aic_geometrically_bounded_surface              ISO 10303-507
   aic_geometrically_bounded_wireframe            ISO 10303-510
   aic_manifold_surface                           ISO 10303-509
   aic_mechanical_design_geometric_presentation   ISO 10303-517
   aic_mechanical_design_shaded_presentation      ISO 10303-518
   aic_non_manifold_surface                       ISO 10303-508
   aic_shell_based_wireframe                      ISO 10303-502
   aic_topologically_bounded_surface              ISO 10303-511
   application_context_schema                     ISO 10303-41
   approval_schema                                ISO 10303-41
   configuration_management_schema                ISO 10303-44
   contract_schema                                ISO 10303-41
   date_time_schema                               ISO 10303-41
```

document_schema	ISO 10303-41
drawing_definition_schema	ISO 10303-101
effectivity_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometry_schema	ISO 10303-42
group_schema	ISO 10303-41
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45
measure_schema	ISO 10303-41
method_definition_schema	ISO 10303-49
person_organization_schema	ISO 10303-41
presentation_appearance_schema	ISO 10303-46
presentation_definition_schema	ISO 10303-46
presentation_organization_schema	ISO 10303-46
presentation_resource_schema	ISO 10303-46
process_property_schema	ISO 10303-49
process_property_representation_schema	ISO 10303-49
product_concept_schema	ISO 10303-44
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41

5.2.1 Fundamental concepts and assumptions

This subclause describes the fundamental concepts and assumptions related to the data structure within this part of ISO 10303. The fundamental concepts and assumptions will be put in the perspective of a collection of generic functional modules to facilitate comprehension of the subject matter. These modules will then be grouped together to form Units of Functionality. Conformance Classes within this part of ISO 10303 typically follow the identified Units of Functionality. Detail descriptions of capabilities for conformance classes 1 through 7 are located in Annex K of this part of 10303.

The following is a matching of Conformance Classes with Units of Functionality.

Conformance Class	**Units of Functionality**
1 - Data Definition Exchange for files	Data Definition Exchange
2 - Data Definition Exchange for TDP elements	Data Definition Exchange
3 - Data Definition Exchange for indentured methods	Data Definition Exchange
4 - Parts List	Parts List
5 - Data List	Data List
6 - Indentured Data List	Indentured Data List
7 - Index List	Index List
8 - Other list	Other list
9 - List with presentation	Presentation (also one of the following: DDE, PL, DL, IDL, IL, OL)
10 - Reference document identification and drawing	Reference Document and Drawing
11 - Reference document identification and drawing w/ ISO 10303-201 and ISO 10303-202	Reference Document and Drawing w/ ISO 10303-201 and ISO 10303-202
12 - Product Data Set without presentation format	Product Data Set

13 - Product Data Set with Shading	Product Data Set
14 - Product Data Set with presentation format	Product Data Set

First, the concept of how product structure and document structure can co-exist will be discussed. Then assumptions and concept of each conformance class will be examined.

5.2.1.1 Product structure and document structure interaction

ISO 10303 is founded on the basis of sharing/exchanging product model data. This part of ISO 10303 is focused on grouping the product data per the product (for example, part or assembly) that is defined. This grouping provides the means for configuration control of the product data from a product structure view point. There are many systems and applications that group and configuration control product data through a non product structure view, that this part of ISO 10303 calls a document view. A document may contain product data for one or more products. This part of ISO 10303 will provide the structure that will allow documents to be identified and configuration controlled in the context of a product structure view point, plus provide the structure so that the contents of a particular set of documents can coexist in a ISO 10303 architecture. These documents include parts list, data list, index list, indentured data list, other lists, and product data sets.

A fundamental concept to this part of ISO 10303 is that the data needed to support configuration control of both documents and products concurrently will be captured. Configuration control will be done for both documents and products through **product**, **product_definition_formation**, **product_definition**, **product_definition_relationship**, **property_definition**, and the management resource constructs in ISO 10303-41. The way to distinguish a part from a document will be done through the **product_category**. For documents, there shall exist a **product_category.name** equal to 'document'. For parts, there shall NOT be an associated **product_category.name** equal to 'document'.

The versioning of a document will be satisfied by instancing a group of related entities. The entities consist of :

product_category.name ='document'
product
product_definition_formation

The **product** and **product_definition_formation** entities provide the versioning capability. The **document_product_equivalence** entity establishes the equivalence between the following:

— **document** entity and the **product_definition_formation**,

— **document** entity and the **product** entity,

— **document** entity and the **product_definition** entity.

The **document** entity provides the links to the document specific functionality like **document_usage_constraint** and **document_reference**.

These specializations and constraints are used for consistent mapping of documents.

Document_product_equivalence is a new entity is created in this part of ISO 10303. This new entity will establish that an instance of a **product**, **product_definition_formation** or a **product_definition** entity that is equivalent to a **document** entity. The new entity will be called "**document_product_equivalence**" and is a subtype of **document_product_association**. The attribute **document_product_equivalence.name** will be constrained to the value of 'equivalence'. Also the value of the attribute **document_type.product_-data_type** will be constrained to 'configuration controlled document' or 'configuration controlled document version' or 'configuration controlled document definition' based on whether **document_product_-equivalence.related_product** reference **product**, **product_definition_formation** or **product_definition**. These constraints are captured as WHERE rules in the definition of **document_product_equivalence** (see 5.2.3.1.25)

Interrelationships among products and documents is another fundamental concept that this part of ISO 10303 supports. **Product_definition_relationship**, and **document_reference** will provide the basic product/document relationships. The **product_definition_relationship** will provide a relationship that may put a document or a part in a subordinate view to each other based on which **product_definition** is related through the attribute **product_definition_relationship.related_product_definition**. The **document_-reference** primarily puts the document in a subordinate view to the related part when the **document_-product_equivalence** path is to a **product_definition_formation**. The **document_reference** is also used as a listing mechanism when the **document_product_equivalence** path is to a **product_definition**.

The string value of the attribute **document_reference_role.name** is the key to the meaning of many relationships associated with document. These string values are denoted in the mapping tables (see 5.1), and the application interpreted model rules (see 5.2.4).

The use of **property_definition** to collect document configuration data is a notable fundamental concept. **Property_definition** plays the role here in the context of a document header information. The attribute **property_definition.name** will contain a string value with the characters 'header' somewhere inside. Each **property_definition** used in this document context will have it's name attribute defined explicitly in the mapping tables (see 5.1), and the application interpreted model rules (see 5.2.4).

Uniquely specifying a document with the use of **product_category** will provide the key to interpreting the context and mappings of certain documents. The attribute **product_category.name** string types will be uniquely denoted in the mapping table (see 5.1), and the application interpreted model rules (see 5.2.4).

5.2.1.2 Data Definition Exchange assumptions

The concept of facilitating a configuration controlled exchange among Product Data Management (PDM) systems is captured in the Data Definition Exchange Unit of Functionality / Conformance Class.

Definition of a Data Definition Exchange is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that document, with the attribute **product_category.name** equal to 'data definition exchange'.

Unique instances of **product_definition_formation** signify content changes within a document. This is done by using **product_definition_formation.id** to identify the unique version based on contents changes and additional information consisting of document revision, change, issue, and sequence identification and their respective dates captured through **executed_action**s.

A unique instance of **product_definition** may signify different external (outside of ISO 10303) representations of a document version. When this is the case, an instance of **product_definition_with_-associated_documents** that is a subtype of **product_definition** is used to identify the unique external representation of the document. A document version (**product_definition_formation**) may have zero, one or many document representations (**product_definition**) associated to itself. Each **product_definition_-with_associated_documents** instance for a document will act as the collector for a set of files (**document_-files**) that contain the content or presentation of a particular version of a document. A **product_definition_-with_associated_documents** may have one or more files (**document_file**) associated to itself.

An external file is captured through the entity **document_file**. **Document_file** is a subtype of **document** and **characterized_object**. With a file being a subtype of **characterized_object**, properties can be associated through the use of **property_definition**. The format of a file is captured through an instance of **representation**. Files (**document_file**) can also be associated to a part (**product**) through the use of a **product_definition_with_associated_documents**, **applied_document_reference**, or **property_definition** for a part's shape.

The concept of interrelating products and documents is a major function of the Data Definition Exchange capability. The entity **applied_document_reference** with the select type **document_reference_item** and associated **object_role.name** equal to 'indentured by document' or 'indentured by item' or 'indentured by item and document' provide the basic mechanism for identifying the group of relationships among documents and products. The associated **object_role.name** provides the type of relationships that may be present in the group. These relationships are of four different types. They are the **product_definition_-relationship**, **product_definition_with_associated_documents**, **applied_document_reference**, and **product_definition_formation_relationship**.

5.2.1.3 Parts List

Definition of a Parts List is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'parts list'. An equivalent instance of the **document** entity is needed if an **applied_-document_reference** or **applied_document_usage_constraint** is to be instantiated . A **document_-product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_type.product_-data_type** equal to 'configuration controlled document version'.

A Parts_list can contain one or more part tabulations. Part Tabulation is a list of parts based on a business need. The primary use is to identify what collection of parts goes into an assembly. A part tabulation can identify what part or stock was used to make a detail part. A part tabulation can also provide a list of parts that make a repair kit. A part tabulation is an Item_list that is mapped to a **product_definition**. This **product_definition** is the definition of an Item or a Tdp_element (document) based on what type of part tabulation is being created. The string value that identifies this **product_definition** as a part tabulation is **product_definition_context.name**='item list'. This context is placed on the **product_definition** through **product_definition_context_association**.

There is another context that is required to be placed on the **product_definition**. This context is placed on the **product_definition** through **product_definition.frame_of_reference**. The string value for this second **product_definition_context.name** attribute provides a distinction on the type of view on a part version ('part definition') from one of a document version ('digital document definition', 'physical document

definition'). This attribute may also indicate other types of definitions: for example, functional, or spatial and/or zonal.

For **product_definition** with a **product_definition_context.name**='item list', another **product_definition_-context.name** is required that provides additional context, defining the type of list. This additional context is place on the **product_definition** through **product_definition_context_association**. The enumeration component_list_type_enumeration defined in section 4.2.10.3 provides examples of some commonly used context for these types of list.

If a **product_definition** is representing an assembly, then the context of **product_definition_-context.name**='assembly definition' is placed on the **product_definition** through **product_definition_-context_association**.

5.2.1.4 Data List

Definition of a Data List is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'data list'. An equivalent instance of the **document** entity is needed if an **applied_-document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_-product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_type.product_-data_type** equal to 'configuration controlled document version'.

5.2.1.5 Indentured Data List

Definition of an Indentured Data List is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_category.name** equal to 'indentured data list'. An equivalent instance of the **document** entity is needed if an **applied_document_reference** or **applied_document_usage_constraint** is to be instantiated.. A **document_product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_-type.product_data_type** equal to 'configuration controlled document version'.

5.2.1.6 Index List

Definition of a Index List is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'index list'. An equivalent instance of the **document** entity is needed if an **applied_document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_-type.product_data_type** equal to 'configuration controlled document version'.

5.2.1.7 Drawing

Definition of a Drawing is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'drawing'. An equivalent instance of the **document** entity is needed if an **applied_-document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_-product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_type.product_-data_type** equal to 'configuration controlled document version'.

5.2.1.8 Product Data Set

Definition of a Product Data Set is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'product data set'. An equivalent instance of the **document** entity is needed if an **applied_document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_-type.product_data_type** equal to 'configuration controlled document version'.

5.2.1.9 Reference Document

Definition of a Reference Document is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'reference document'. An equivalent instance of the **document** entity is needed if an **applied_document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_-type.product_data_type** equal to 'configuration controlled document version'.

5.2.1.10 Other List

Definition of an Other List is through identification of a **product_definition_formation**, a **product** for that **product_definition_formation**, a **product_category** for that **product**, with an attribute **product_-category.name** equal to 'other list'. An equivalent instance of the **document** entity is needed if an **applied_-document_reference** or **applied_document_usage_constraint** is to be instantiated. A **document_-product_equivalence** instance is created referencing the **product_definition_formation** and **document** instance. The **document** instance references a **document_type** with the attribute **document_type.product_-data_type** equal to 'configuration controlled document version'.

5.2.1.11 List Presentation

Definition of a List Presentation is through identification of a **presentation_set** with a **presented_item** identifying the document it is presenting.

5.2.2 Technical_data_packaging types

5.2.2.1 Technical_data_packaging constant and type definitions

5.2.2.1.1 action_item

An **action_item** specifies a **configuration_effectivity**, **document_file**, **product_definition**, **product_definition_with_associated_documents**, **product_definition_formation**, **product_definition_relationship**, or **representation** that an **executed_action** may be assigned.

EXPRESS specification:

```
*)
TYPE action_item = SELECT
  (configuration_effectivity,
   document_file,
   product_definition,
   product_definition_with_associated_documents,
   product_definition_relationship,
   product_definition_formation,
   representation);
END_TYPE;
(*
```

5.2.2.1.2 action_request_item

An **action_request_item** specifies an **product_definition**, **product_definition_formation**, **product_definition_relationship**, or **property_definition** which a **versioned_action_request** may be assigned.

EXPRESS specification:

```
*)
TYPE action_request_item = SELECT
  (product_definition,
   product_definition_formation,
   product_definition_relationship,
   property_definition);
END_TYPE;
(*
```

5.2.2.1.3 approval_item

An **approval_item** specifies an **action**, **applied_action_assignment**, **applied_certification_assignment**, **applied_effectivity_assignment**, **certification**, **configuration_effectivity**, **configuration_item**, **contract**, **document_file**, **product_definition**, **product_definition_formation**, **product_definition_relationship**, **property_definition**, or **versioned_action_request** which an **approval** may be assigned.

EXPRESS specification:

```
*)
TYPE approval_item = SELECT
  (action,
   applied_action_assignment,
   applied_certification_assignment,
   applied_effectivity_assignment,
   certification,
   configuration_effectivity,
   configuration_item,
   contract,
   document_file,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   property_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.4 certification_item

A **certification_item** specifies a **product_definition_formation_relationship** which a **certification** may be assigned.

EXPRESS specification:

```
*)
TYPE certification_item = SELECT
  (product_definition_formation_relationship);
END_TYPE;
(*
```

5.2.2.1.5 classification_item

A **classification_item** specifies a **action_status**, **approval**, **descriptive_representation_item**, or **product_-definition** which a **class or class_system** may be assigned.

EXPRESS specification:

```
*)
TYPE classification_item = SELECT
  (action_status,
   approval,
   descriptive_representation_item,
   product_definition);
END_TYPE;
(*
```

5.2.2.1.6 contract_item

A **contract_item** specifies a **action**, **product_definition**, **product_definition_formation**, **product_-definition_with_associated_documents**, or **property_definition** which a **contract** may be assigned.

EXPRESS specification:

```
*)
TYPE contract_item = SELECT
  (action,
   product_definition,
   product_definition_formation,
   product_definition_with_associated_documents,
   property_definition);
END_TYPE;
(*
```

5.2.2.1.7 date_and_time_item

A **date_and_time_item** specifies an **action**, **applied_action_assignment**, **applied_contract_assignment**, **applied_security_classification_assignment, approval_person_organization**, **certification**, **contract**, **event_occurrence**, **organizational_project**, **product_definition**, **product_definition_formation**, **representation**, **security_classification**, or **versioned_action_request** which a **date_and_time** may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_time_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   event_occurrence,
   organizational_project,
   product_definition,
   product_definition_formation,
   representation,
   security_classification,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.8 date_item

A **date_item** specifies an **action**, **applied_action_assignment**, **applied_contract_assignment**, **applied_-security_classification_assignment**, **approval_person_organization**, **certification**, **contract**, **event_-occurrence**, **organizational_project**, **product_definition**, **product_definition_formation**, **representation**, **security_classification**, or **versioned_action_request** which a **date** may be assigned.

EXPRESS specification:

```
*)
TYPE date_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   event_occurrence,
   organizational_project,
   product_definition,
   product_definition_formation,
   representation,
   security_classification,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.9 document_reference_item

A **document_reference_item** specifies an **action**, **action_method**, **alternate_product_relationship**, **applied_document_reference**, **assembly_component_usage_substitute_with_ranking**, **descriptive_representation_item**, **document_file**, **executed_action**, **externally_defined_symbol_and_placement**, **make_from_usage_option**, **make_from_usage_option_with_reference_designator**, **next_assembly_usage_occurrence**, **product**, **product_definition**, **product_definition_formation**, **product_definition_formation_relationship**, **product_definition_occurrence_relationship**, **product_definition_relationship**, **product_definition_with_associated_documents**, **promissory_usage_occurrence**, **property_definition**, **quantified_assembly_component_usage**, **shape_aspect**, **shape_aspect_relationship**, **specified_higher_usage_occurrence** or **versioned_action_request** which a **document** may be assigned.

EXPRESS specification:

```
*)
TYPE document_reference_item = SELECT
  (action,
   action_method,
   alternate_product_relationship,
   applied_document_reference,
   assembly_component_usage_substitute,
   descriptive_representation_item,
   document_file,
   document_relationship,
   executed_action,
   externally_defined_symbol_and_placement,
   make_from_usage_option,
   make_from_usage_option_with_reference_designator,
   next_assembly_usage_occurrence,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_occurrence_relationship,
```

```
   product_definition_relationship,
   product_definition_with_associated_documents,
   promissory_usage_occurrence,
   property_definition,
   quantified_assembly_component_usage,
   shape_aspect,
   shape_aspect_relationship,
   specified_higher_usage_occurrence,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.10 effectivity_context_item

A **effectivity_context_item** specifies an **organization** or **product_definition_formation** that provides the **context** to an **effectivity_assignment.**

EXPRESS specification:

```
*)
TYPE effectivity_context_item  = SELECT
  (organization,
   product_definition_formation);
END_TYPE;
(*
```

5.2.2.1.11 effectivity_item

A **effectivity_item** specifies a **product_definition**, **product_definition_formation**, or **product_definition_-relationship** which an **effectivity** is being placed upon.

EXPRESS specification:

```
*)
TYPE effectivity_item  = SELECT
  (product_definition,
   product_definition_formation,
   product_definition_relationship);
END_TYPE;
(*
```

5.2.2.1.12 event_occurrence_item

An **event_occurrence_item** specifies an **organizational_project** which an **event_occurrence** may be assigned.

EXPRESS specification:

```
*)
TYPE event_occurrence_item  = SELECT
  (organizational_project);
END_TYPE;
(*
```


5.2.2.1.13 external_identification_item

An **external_identification_item** specifies an additional identification to a File or location of product representation. The identification is external to this part of ISO 10303.

EXPRESS specification:

```
*)
TYPE external_identification_item  = SELECT
  (document_file,
   product_definition);
END_TYPE;
(*
```

5.2.2.1.14 identification_item

A **identification_item** specifies an **application_context**, **applied_document_reference**, **applied_-document_usge_constraint_assignment**, **approval_status**, **document_file**, **organization**, **product, product_definition**, **product_definition_formaiton**, **security_classification_level**, or **shape_aspect_-relationship** that an identifying string is assigned.

NOTE The identifying string may be a version identification for a **document_file**. The identifying string may be a company code for an **organization**.

EXPRESS specification:

```
*)
TYPE identification_item  = SELECT
  (application_context,
   applied_document_reference,
   applied_document_usage_constraint_assignment,
   approval_status,
   document_file,
   organization,
   product,
   product_definition,
   product_definition_formation,
   security_classification_level,
   shape_aspect_relationship);
END_TYPE;
(*
```

5.2.2.1.15 language_item

A **language_item** specifies a **representation** that an identifying string is assigned

EXPRESS specification:

```
*)
TYPE language_item = SELECT
   (representation);
END_TYPE;
(*
```

5.2.2.1.16 organization_item

An **organization_item** specifies an **action**, **applied_action_assignment**, **applied_contract_assignment**, **applied_identification_assignment**, **configuration_item**, **contract**, **document_file**, **event_occurrence**, **organizational_project**, **product**, **product_definition**, **product_definition_formation**, **security_-classification**, or **versioned_action_request** which an **organization** may be assigned.

EXPRESS specification:

```
*)
TYPE organization_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_identification_assignment,
   configuration_item,
   contract,
   document_file,
   event_occurrence,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   security_classification,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.17 organizational_project_item

An **organizational_project_item** specifies a **product_concept** which an **organizational_project** may be assigned.

EXPRESS Specification:

```
*)
TYPE organizational_project_item = SELECT
   (action, product_concept);
END_TYPE;
(*
```


5.2.2.1.18 person_and_organization_item

A **person_and_organization_item** specifies an **action**, **applied_action_assignment**, **applied_contract_-assignment**, **applied_identification_assignment**, **configuration_item**, **contract**, **document_file**, **event_-occurrence**, **organizational_project**, **product**, **product_definition**, **product_definition_formation**, **security_classification**, or **versioned_action_request** which a **person_and_organization** may be assigned.

NOTE A **product_definition_formation** may be either a version of a part or a version of a document to which a person and organization may be assigned.

EXPRESS specification:

```
*)
TYPE person_and_organization_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_identification_assignment,
   configuration_item,
   contract,
   document_file,
   event_occurrence,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   security_classification,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.1.19 presented_item_item

A **presented_item_item** specifies the version of a document which a **presented_item** may be assigned.

EXPRESS specification:

```
*)
TYPE presented_item_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.1.20 security_classification_item

A **security_classification_item** specifies a **document_file**, **product_definition**, **product_definition_-formation**, **product_definition_relationship** or **property_definition** which a **security_classification** may be assigned.

EXPRESS specification:

```
*)
TYPE security_classification_item  = SELECT
  (document_file,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   property_definition);
END_TYPE;
(*
```

5.2.3 Technical_data_packaging entities

5.2.3.1 Technical_data_packaging entity definitions

5.2.3.1.1 applied_action_assignment

An **applied_action_assignment** is a type of **action_assignment** that identifies the action taken against a document or product. The action is related to either the version or the status of the document or product. The document or product is in a list that identifies the **document** or **product** by the revision, change, and issue.

EXPRESS specification:

```
*)
ENTITY applied_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **action_item**s which identifies the **configuration_effectivity**, **document_file**, **product_definition**, **product_definition_with_associated_documents**, **product_definition_formation**, **product_definition_relationship**, or **representation** that the **applied_action_assignment** applies.

5.2.3.1.2 applied_action_request_assignment

An **applied_action_request_assignment** is a type of **action_request_assignment** that identifies the versioned_action_request taken against a document or product. The **versioned_action_request** is related to either the version or the status of the document or product. The document or product is in a list that identifies the document or product by the revision, change, and issue.

EXPRESS specification:

```
*)
ENTITY applied_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF action_request_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **action_request_item**s that identifies the **product_definition**, **product_definition_formation**, **product_definition_relationship**, or **property_definition** that the **applied_action_request_assignment** applies.

5.2.3.1.3 applied_approval_assignment

An **applied_approval_assignment** is a type of **approval_assignment** that specifies an **approval** for an **action**, **applied_action_assignment**, **applied_certification_assignment**, **applied_effectivity_assignment**, **certification**, **configuration_effectivity**, **configuration_item**, **contract**, **document_file**, **product_definition**, **product_definition_formation**, **product_definition_relationship**, **property_definition**, or **versioned_action_request**

EXPRESS specification:

```
*)
ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approval_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **approval_item**s that identify the particular product data that the **approval** applies.

5.2.3.1.4 applied_certification_assignment

An **applied_certification_assignment** is a type of **certificatoin_assignment** that assigns a **certification** to a particular piece of product data .

EXPRESS specification:

```
*)
ENTITY applied_certification_assignment
  SUBTYPE OF (certification_assignment);
  items : SET [1:?] OF certification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **certification_item**s that identify the particular **product_definition_formation_relationship** that the **certification** applies.

5.2.3.1.5 applied_classification_assignment

An **applied_classification_assignment** is a type of **classification_assignment** that assigns a **class** or **class_system** to a particular piece of product data .

EXPRESS specification:

```
*)
ENTITY applied_classification_assignment
  SUBTYPE OF (classification_assignment);
  items : SET [1:?] OF classification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **classification_item**s that identify the particular **action_status**, **approval**, **descriptive_representation_item**, or **product_definition** that the **class** or **class_system** applies.

5.2.3.1.6 applied_contract_assignment

An **applied_contract_assignment** is a type of **contract_assignment** that assigns a **contract** to a particular piece of product data .

EXPRESS specification:

```
*)
ENTITY applied_contract_assignment
  SUBTYPE OF (contract_assignment);
  items : SET [1:?] OF contract_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **contract_item**s that identify the particular **action**, **product_definition**, **product_definition_formation**, **product_definition_with_associated_documents**, or **property_definition** that the **contract** applies.

5.2.3.1.7 applied_date_and_time_assignment

An **applied_date_and_time_assignment** is a type of **date_and_time_assignment** that specifies the **date_and_time** to a particular piece of product data.

EXPRESS specification:

```
*)
ENTITY applied_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET [1:?] OF date_and_time_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **date_and_time_item**s that identify the particular **action**, **applied_action_assignment**, **applied_contract_assignment**, **applied_security_classification_assignment,** **approval_person_-organization**, **certification**, **contract**, **event_occurrence**, **organizational_project**, **product_definition**, **product_definition_formation**, **representation**, **security_classification**, or **versioned_action_request** that the **date_and_time** applies.

5.2.3.1.8 applied_date_assignment

An **applied_date_assignment** is a type of **date_assignment** that specifies the **date** for a particualr piece of product data.

EXPRESS specification:

```
*)
ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **date_item**s that identify the particular **action**, **applied_action_assignment**, **applied_-contract_assignment**, **applied_security_classification_assignment**, **approval_person_organization**, **certification**, **contract**, **event_occurrence**, **organizational_project**, **product_definition**, **product_-definition_formation**, **representation**, **security_classification**, or **versioned_action_request** that the **date** applies.

5.2.3.1.9 applied_document_reference

An **applied_document_reference** is a type of **document_reference** that specifies the relationship between a **document** and a particular piece of product data.

An **applied_document_reference** assigns a **product_definition_with_associated_documents**, **product_-definition_formation_relationship**, **product_definition_relationship**, or **applied_document_reference** as a member of a collection of parent/child relationship among parts and parts, documents and documents,parts and documents, and documents and parts. An **applied_document_reference** may assign a **product_definition_formation** as a member of a collection of documents. An **applied_document_-reference** may assign a **document_file** as a member of a collection of electronic data files.

NOTE 1 The **applied_document_reference** with the **document_reference_item** provides the structure to relate products and documents together from multiple contexts.

NOTE 2 An **applied_document_reference** will be used in both a Data Definition Exchange capability and a Indentured Data List capability.

NOTE 3 The **document_role** that is associated with the **applied_document_reference** specifies the context of the relationship the **applied_document_reference** establishes.

EXPRESS specification:

```
*)
ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_reference_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **document_reference_item**s that identify the particular **action**, **action_method**, **alternate_product_relationship**, **applied_document_reference**, **assembly_component_usage_substitute**, **descriptive_representation_item**, **document_file**, **externally_defined_symbol_and_placement**, **make_from_usage_option**, **make_from_usage_option_with_reference_designator**, **next_assembly_usage_occurence**, **product**, **product_definition**, **product_definition_formation**, **product_definition_formation_relationship**, **product_definition_occurrence_relationship, product_definition_relationship**, **product_definition_with_associated_documents**, **promissory_usage_occurrence, property_definition**, **quantified_assembly_component_usage**, **shape_aspect**, **shape_aspect_relationship**, **specified_higher_usage_occurrence**, or **versioned_action_request** that the **document_reference** applies.

5.2.3.1.10 applied_document_usage_constraint_assignment

An **applied_document_usage_constraint** is a type of **document_usage_constraint** that specifies the relationship between a **document** and a particular piece of product data. Within the relationship there is a constraint identified on the used of the document.

NOTE The **document_usage_constraint_role** that is associated with the **applied_document_usage_constraint** specifies the context of the relationship the **applied_document_usage_constraint** establishes.

EXPRESS specification:

```
*)
ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
  items : SET [1:?] OF document_reference_item;
END_ENTITY;
(*
```


Attribute definitions:

items: the set of **document_reference_item**s that identify the particular **action**, **action_method**, **alternate_product_relationship**, **applied_document_reference**, **assembly_component_usage_substitute**, **descriptive_representation_item**, **make_from_usage_option**, **make_from_usage_option_with_reference_designator**, **next_assembly_usage_occurence**, **product**, **product_definition**, **product_definition_formation**, **product_definition_formation_relationship**, **product_definition_occurrence_relationship**, **product_definition_relationship**, **product_definition_with_associated_documents**, **promissory_usage_occurrence**, **property_definition**, **quantified_assembly_component_usage**, **shape_aspect**, **shape_aspect_relationship**, **specified_higher_usage_occurrence**, or **versioned_action_request** that the **document_usage_constraint** applies.

5.2.3.1.11 applied_effectivity_assignment

An **applied_effectivity_assignment** is a type of **effectivity_assignment** that specifies the relationship between an **effectivity** and a particular piece of product data.

EXPRESS specification:

```
*)
ENTITY applied_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
  items : SET [1:?] OF effectivity_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **effectivity_item**s that identify the particular **product_definition**, **product_definition_formation**, or **product_definition_relationship** that the **effectivity** applies.

5.2.3.1.12 applied_effectivity_context_assignment

An **applied_effectivity_context_assignment** is a type of **effectivity_context_assignment** that specifies the relationship between a **applied_effectivity_assignment** and a particular piece of product data defining the context of the **applied_effectivity_assignment**.

EXPRESS specification:

```
*)
ENTITY applied_effectivity_context_assignment
  SUBTYPE OF (effectivity_context_assignment);
  items : SET [1:?] OF effectivity_context_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **effectivity_context_item**s that identify the particular **organization** or **product_definition_formation** that the **effectivity_context** applies.

5.2.3.1.13 applied_event_occurrence_assignment

An **applied_event_occurrence_assignment** is a type of **event_occurrence_assignment** that specifies the event_occurrence for a particular piece of product.

EXPRESS specification:

```
*)
ENTITY applied_event_occurrence_assignment
  SUBTYPE OF (event_occurrence_assignment);
  items : SET [1:?] OF event_occurrence_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **event_occurrence_item**s that identify the particular **organizational_project** that the **event_occurrence** applies.

5.2.3.1.14 applied_external_identification_assignment

An **applied_external_identification_assignment** is a type of **external_identification_assignment** that specifies the identification of a particular piece of product data at an **external_source**.

EXPRESS specification:

```
*)
ENTITY applied_external_identification_assignment
  SUBTYPE OF (external_identification_assignment);
  items : SET [1:?] OF external_identification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **external_identification_item**s that identify the particular **document_file**, **product_-definition,** or **product_definition_with_associated_documents** that the **external_identification** applies.

5.2.3.1.15 applied_identification_assignment

An **applied_identification_assignment** is a type of **identification_assignment** that specifies the relationship between an **identification_assignment** and a particular piece of product data.

NOTE An **applied_identification_assignment** specifies the version of a **document_file** or the alias for the **document_file** id. An **applied identification_assignment** can also specify the company code for an **organization**.

EXPRESS specification:

```
*)
ENTITY applied_identification_assignment
  SUBTYPE OF (identification_assignment);
  items : SET [1:?] OF identification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **idenfication_item**s that identify the particular **application_context**, **applied_document_-reference**, **applied_document_usge_constraint_assignment**, **approval_status**, **document_file**, **organization**, **product, product_definition**, **product_definition_formaiton**, **security_classification_level**, or **shape_aspect_relationship** that the identifier applies.

5.2.3.1.16 applied_organization_assignment

An **applied_organization_assignment** is a type of **organization_assignment** that specifies the **organization** for a particular piece of product data..

EXPRESS specification:

```
*)
ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **organization_item**s that identify the particular **action**, **applied_action_assignment**, **applied_contract_assignment**, **applied_identification_assignment**, **configuration_item**, **contract**, **document_file**, **event_occurrence**, **organizational_project**, **product**, **product_definition**, **product_-definition_formation**, **security_classification**, or **versioned_action_request** that the **organization** applies.

5.2.3.1.17 applied_organizational_project_assignment

An **applied_organizational_project_assignment** is a type of **organizational_project_assignment** that specifies the **organizational_project** for a particular piece of product data.

EXPRESS Specification:

```
*)
ENTITY applied_organizational_project_assignment
  SUBTYPE OF (organizational_project_assignment);
  items : SET [1:?] OF organizational_project_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **organizational_project_items** that identify the particular **action** or **product_concept** that the **organizational_project** applies.

5.2.3.1.18 applied_person_and_organization_assigmnent

An **applied_persion_and_organization_assignment** is a type of **person_and_organization_assignment** that specifies the **person and organization** for a particular piece of product data.

EXPRESS specification:

```
*)
ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **person_and_ organization_item**s that identify the particular **action**, **applied_action_-assignment**, **applied_contract_assignment**, **applied_identification_assignment**, **configuration_item**, **contract**, **document_file**, **event_occurrence**, **organizational_project**, **product**, **product_definition**, **product_definition_formation**, **security_classification**, or **versioned_action_request** that the **person_-and_organization** applies.

5.2.3.1.19 applied_presented_item

A **applied_presented_item** is a type of **presented_item** that identifies a **presented_item** to a particular **product_definition_formation**.

EXPRESS specification:

```
*)
ENTITY applied_presented_item
  SUBTYPE OF (presented_item);
  items : SET [1:?] OF presented_item_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **presented_item_item** that identify the particular **product_definition_formation** that the **presented_item** applies.

5.2.3.1.20 applied_security_classification_assignment

An **applied_security_classification_assignment** is a type of **security_classification_assignment** that specifies the **security_classification** for a particular piece of product data.

EXPRESS specification:

```
*)
ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **security_classification_item**s that identify the particular **document_file**, **product_definition**, **product_definition_formation**, **product_definition_relationship** or **property_definition** that the **security_classification** applies.

5.2.3.1.21 class

A **class** is a type of **group** that is a classification of an object which characterizes all objects of the same kind; such a classification is independent from the application of the classified object..

EXPRESS specification:

```
*)
ENTITY class
  SUBTYPE OF (group);
END_ENTITY;
(*
```

5.2.3.1.22 class_system

A **class_system** is a type of **group** that specifies a method for classifing.

EXPRESS specification:

```
*)
ENTITY class_system
  SUBTYPE OF (group);
END_ENTITY;
(*
```

5.2.3.1.23 design_make_from_relationship

A **design_make_from_relationship** is a type of **product_definition_relationship** that specifies that one product's design has been derived from another product's design. The use of the **design_make_from_relationship** also implies that the physical product resulting from the manufacture of the **product_definition_relationship.relating_product_definition** is to be used as the basis for the manufacture of the **product_definition_relationship.related_product_definition**.

EXPRESS specification:

```
*)
ENTITY design_make_from_relationship
  SUBTYPE OF (product_definition_relationship);
END_ENTITY;
(*
```

Attribute definitions:

SELF/product_definition_relationship.relating_product_definition: the **product_definition** that is the source of the make from relationship.

SELF/product_definition_relationship.related_product_definition: the **product_definition** entity that is created from the source product_definition.

NOTE The above definition, EXPRESS specification, and attribute definition is from ISO 10303-203:1994.

5.2.3.1.24 document_file

A **document_file** is a type of **document** and a type of **characterized_object** that is a file on a computer system that represents the whole or a portion of a document.

NOTE One or more **document_file**s may make up a complete representation of a document.

EXPRESS specification:

```
*)
ENTITY document_file
   SUBTYPE OF (document,
     characterized_object);
WHERE
  WR1: (SELF\characterized_object.name = '');
  WR2: (NOT EXISTS(SELF\characterized_object.description));
  WR3: (SIZEOF (QUERY( drt <* SELF\document.representation_types |
(drt.name IN ['digital','physical']) )) = 1);
END_ENTITY;
(*
```

Formal propositions:

WR1: inherited attribute **characterized_object.name** in **document_file** entity shall be blank.

WR2: inherted attribute **characterized_object.description** in **document_file** entity shall be blank.

WR3: The document shall have exactly one document representation type with a name of either 'digital' or 'physical'.

5.2.3.1.25 document_product_equivalence

A **document_product_equivalence** is a type of **document_product_association** that establishes a one to one corespondence between a **product_definition_formation**, **product_definition,** or **product** construct and a **document** construct for identifying the concept of a single document.

NOTE A **document_product_equivalence** ties the functionality in both the ISO 10303-41 **document_schema** and the **product_definition_schema** so that document accountability can be satisfied.

EXPRESS specification:

```
*)
ENTITY document_product_equivalence
   SUBTYPE OF (document_product_association);
WHERE
WR1: SELF.name = 'equivalence';
WR2: NOT('TECHNICAL_DATA_PACKAGING.PRODUCT'
IN TYPEOF(SELF.related_product)) OR
((SELF.relating_document.kind.
product_data_type = 'configuration controlled document') AND
(SIZEOF( QUERY(
prpc <* USEDIN(SELF.related_product,'TECHNICAL_DATA_PACKAGING.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
prpc.name = 'document')) = 1));
WR3: NOT('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION' IN
TYPEOF(SELF.related_product)) OR
((SELF.relating_document.kind.product_data_type =
'configuration controlled document version') AND
(SIZEOF( QUERY( prpc <* USEDIN(SELF.related_product\
product_definition_formation.of_product,
'TECHNICAL_DATA_PACKAGING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
prpc.name = 'document')) = 1));
WR4: NOT('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION' IN
TYPEOF(SELF.related_product)) OR
((SELF.relating_document.kind.product_data_type =
'configuration controlled document definition') AND
(SIZEOF( QUERY( prpc <* USEDIN(SELF.related_product\
product_definition.formation.of_product,
'TECHNICAL_DATA_PACKAGING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
prpc.name = 'document'
)) = 1));
END_ENTITY;
(*
```

Formal propositions:

WR1: **document_product_equivalence.name** shall equal 'equivalence'.

WR2: If **document_product_equivalence.related_product** references **product**, THEN **document_-product_equivalence.relating_document** shall reference a **document** WHERE the **document.kind** shall reference a **document_type** WHERE the **document_type.product_data_type** shall equal 'configuration controlled document' AND there is an associated **product_category** with **product_-category.name** equal to 'document'.

WR3: If **document_product_equivalence.related_product** references **product_definition_formation**, THEN **document_product_equivalence.relating_document** shall reference a **document** WHERE the **document.kind** shall reference a **document_type** WHERE the **document_type.product_data_type** shall equal ‘configuration controlled document version', AND there is an associated **product_category** with **product_category.name** equal to ‘document'.

WR4: If **document_product_equivalence.related_product** references **product_definition**, THEN **document_product_equivalence.relating_document** shall reference a **document** WHERE the **document.kind** shall reference a **document_type** WHERE the **document_type.product_data_type** shall equal ‘configuration controlled document definition', AND there is an associated **product_-category** with **product_category.name** equal to ‘document'.

5.2.3.1.26 externally_defined_class

An **externally_defined_class** is a type of **class** and a type of **externally_defined_item** that represents classification information whose identification and definition is defined externally, that is not within this part of ISO 10303.

EXPRESS specification:

```
*)
ENTITY externally_defined_class
SUBTYPE OF (externally_defined_item,
  class);
END_ENTITY;
(*
```

5.2.3.1.27 externally_defined_general_property

An **externally_defined_general_property** is a type of **general_property** and a type of **externally_-defined_item** that represents general property definition information which is defined externally, that is not within this part of ISO 10303.

EXPRESS specification:

```
*)
ENTITY externally_defined_general_property
SUBTYPE OF (externally_defined_item,
  general_property);
END_ENTITY;
(*
```

5.2.3.1.28 externally_defined_planar_box

An **externally_defined_planar_box** is a type of **planar_box** and a type of **externally_defined_item** with an identified externally defined page size code.

EXPRESS specification:

```
*)
ENTITY externally_defined_planar_box
SUBTYPE OF (externally_defined_item,
  planar_box);
END_ENTITY;
(*
```

5.2.3.1.29 externally_defined_symbol_and_placement

An **externally_defined_symbol_and_placement** is a type of **placement** and a type of **externally_defined_-symbol** with an identified **placement**.

EXPRESS specification:

```
*)
ENTITY externally_defined_symbol_and_placement
SUBTYPE OF (externally_defined_symbol,
  placement);
END_ENTITY;
(*
```

5.2.3.1.30 language_assignment

A **language_assignment** is a type of **classification_assignment**. It specifies a language for the items it is assigned to. This language shall be specified in the **name** of the **group** that is refernced in the role of the **assigned classification**.

EXPRESS specification:

```
*)
ENTITY language_assignment
  SUBTYPE OF (classification_assignment);
 items : SET [1:?] OF language_item;
END_ENTITY;
(*
```

Attribute definitions:

items : the set of **product**s, **product_definition_formation**s, **product_definition**s or **document_file**s to which a language is assigned.

5.2.3.1.31 make_from_usage_option_with_reference_designator

A **make_from_usage_option_with_reference_designator** is a type of **make_from_usage_option** where a reference designator can be captured and where a substitute can be identified through the use of **assembly_component_usage_substitute.**

EXPRESS specification:

```
*)
ENTITY make_from_usage_option_with_reference_designator
 SUBTYPE OF (make_from_usage_option, assembly_component_usage);
END_ENTITY;
(*
```

5.2.3.1.32 other_list_table_representation

An **other_list_table_representation** is a type of **representation** that is a table of information that is used for design disclosure for the TDP element that the **other_list** is associated to.

EXPRESS specification:

```
*)
ENTITY other_list_table_representation
  SUBTYPE OF (representation);
END_ENTITY;
(*
```

5.2.3.2 technical_data_packaging imported entity modifications

5.2.3.2.1 approval

The base definition of the **approval** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval** entity:

— approval_requires_approval_date_time (see 5.2.4.3)

— approval_requires_approval_person_organization (see 5.2.4.4)

— distribution_notice_approval_requires_supporting_data (see 5.2.4.12)

— release_authentication_string_restriction (see 5.2.4.29)

5.2.3.2.2 approval_date_time

The base definition of the **approval_date_time** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval_date_time** entity:

— approval_requires_approval_date_time (see 5.2.4.3)

— reference_document_requires_subcategorization (see 5.2.4.28)

5.2.3.2.3 approval_person_organization

The base definition of the **approval_person_organization** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval_person_organization** entity:

— approval_requires_approval_person_organization (see 5.2.4.4)

— distribution_notice_approval_requires_supporting_data (see 5.2.4.12)

— release_authentication_string_restriction (see 5.2.4.29)

5.2.3.2.4 document

The base definition of the **document** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **document** entity:

— document_product_equivalence_existence_rule (see 5.2.4.13)

5.2.3.2.5 drawing_sheet_revision

The base definition of the **drawing_sheet_revision** entity is given in ISO 10303-101. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **drawing_sheet_revision** entity:

— existence_dependent_drawing_sheet_revision (see 5.2.4.16)

5.2.3.2.6 executed_action

The base definition of the **executed_action** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **executed_action** entity:

— change_identification_restricts_executed_action (see 5.2.4.5)

5.2.3.2.7 named_unit

The base definition of the **named_unit** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **named_unit** entity:

— dependent_instantiable_named_unit (see 5.2.4.11)

5.2.3.2.8 product

The base definition of the **product** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following rules defined in this part of ISO 10303 apply to the **product** entity:

— product_requires_version (see 5.2.4.26)

5.2.3.2.9 product_definition

The base definition of the **product_definition** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition** entity:

— file_format_restricted_to_one_data_definition_entry (see 5.2.4.17)

5.2.3.2.10 product_definition_context_association

The base definition of the **product_definition_context_association** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_context_-association** entity:

— file_format_restricted_to_one_data_definition_entry (see 5.2.4.17)

5.2.3.2.11 product_definition_formation

The base definition of the **product_definition_formation** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_formation** entity:

— data_definition_exchange_restrict_to_one_presentation (see 5.2.4.10)

— product_requires_person_organization (see 5.2.4.27)

— product_requires_version (see 5.2.4.26)

5.2.3.2.12 product_definition_formation_relationship

The base definition of the **product_definition_formation_relationship** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_formation_-relationship** entity:

— data_definition_entry_string_restrict_for_superseded_element (see 5.2.4.8)

5.2.3.2.13 product_definition_relationship

The base definition of the **product_definition_relationship** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_relationship** entity:

— data_definition_entry_string_restrict_for_superseded_element (see 5.2.4.8)

5.2.3.2.14 property_definition

The base definition of the **property_definition** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **property_definition** entity:

— exchange_reason_existence_required (see 5.2.4.15)

— file_format_restricted_to_one_data_definition_entry (see 5.2.4.17)

— header_configuration_restricts_property_definition (see 5.2.4.18)

— item_source_information_identification_constraint (see 5.2.4.23)

5.2.3.2.15 product_related_product_category

The base definition of the **product_related_product_category** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_related_product_-category** entity:

— data_definition_exchange_categorized_as_document (see 5.2.4.9)

— identification_of_sheet_constraint (see 5.2.4.19)

— indentured_data_list_identification_constraint (see 5.2.4.20)

— reference_document_requires_subcategorization (see 5.2.4.28)

5.2.3.2.16 product_relationship

The base definition of the **product_relationship** entity is given in ISO 10303-41. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_relationship** entity:

— drawing_suffix_number_combination_identification_constraint (see 5.2.4.14)

5.2.3.2.17 representation

The base definition of the **representation** entity is given in ISO 10303-43. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **representation** entity:

— exchange_reason_existence_required (see 5.2.4.15)

— indentured_level_tag_identification_constraint (see 5.2.4.21)

— notation_type_identification_constraint (see 5.2.4.24)

5.2.3.2.18 text_literal

The base definition of the **text_literal** entity is given in ISO 10303-46. The following modifications to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **text_literal** entity:

— text_literal_alignment_baseline_constraint (see 5.2.4.33)

— text_literal_font_requires_externally_defined_text_font (see 5.2.4.34)

5.2.4 Technical_data_packaging rule definitions

5.2.4.1 access_file_requires_external_identification

The **access_file_requires_external_identification** rule specifies how a **document_file** being identified for access requires an association with an **external_identification_assignment**.

NOTE 1 This rule applies to mapping table rule 1.

EXPRESS specification:

```
*)
RULE access_file_requires_external_identification
     FOR (document_file);
```

```
LOCAL
  result :  BOOLEAN  := TRUE;
  df     : SET OF document_file;
  aeia   : BAG OF applied_external_identification_assignment;
END_LOCAL;
  df := document_file;
  REPEAT  i := 1 TO SIZEOF (df);
    IF df[i]\document.description = 'access data file' THEN
      aeia := USEDIN(df[i], 'TECHNICAL_DATA_PACKAGING.'+
                    'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS');
      IF (SIZEOF(aeia) < 1) THEN
        result := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

document_file: identifies the set of all instances of **document_file** entities.

Formal propositions:

WR1: A **document_file** where supertype **document.description** ='access data file' shall have an **applied_-external_identification_assignment.items[i**] that points to the **document_file**.

NOTE 2 **Identification_role.name** shall name the specific interpretation context that applies. An example could be 'URL' that stands for Uniform Resource Locator.

5.2.4.2 applied_document_usage_constraint_assignment_restriction

The **applied_document_usage_constraint_assignment_restriction** rule specifies that a **document_file** or an **externally_defined_symbol_and_placement** can not be referenced by an **applied_document_usage_-constraint_assignment**.

NOTE These rule restricts the use of the select type document_reference_item when being referenced by applied_-document_usage_constraint_assignment.

EXPRESS specification:

```
*)
RULE applied_document_usage_constraint_assignment_restriction
        FOR (document_file, externally_defined_symbol_and_placement);
LOCAL
  result : BOOLEAN := TRUE;
  aduca : BAG OF applied_document_usage_constraint_assignment;
  df  :  SET OF document_file;
  exds  :  SET OF externally_defined_symbol_and_placement;
END_LOCAL;
  df := document_file;
  REPEAT i := 1 TO SIZEOF (df);
```



```
    aduca := USEDIN(df[i], 'TECHNICAL_DATA_PACKAGING.'+
                    'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS');
     IF (SIZEOF(aduca ) > 0) THEN
          result := FALSE;
     END_IF;
  END_REPEAT;
  exds := externally_defined_symbol_and_placement;
  REPEAT i := 1 TO SIZEOF (exds);
    aduca := USEDIN(exds[i], 'TECHNICAL_DATA_PACKAGING.'+
                    'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS');
    IF (SIZEOF(aduca ) > 0) THEN
          result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

document_file: identifies the set of all instances of **document_file** entities.

externally_defined_symbol_and_placement : identifies the set of all instances of **externally_defined_-symbol_and_placement**.

Formal propositions:

WR1: The two members, **document_file** and **externally_defined_symbol_and_placement**, of the select type **document_reference_item** should be disallowed when **document_reference_item** is referenced by **applied_document_usage_constraint_assigment.item[i]**.

5.2.4.3 approval_requires_approval_date_time

The **approval_requires_approval_date_time** rule specifies that each instance of **approval** shall be referenced by exactly one **approval_date**. This rule enforces the requirement for every **approval** to have a **date** on which the approval obtained its specified status.

NOTE This rule applies to mapping table rule 2.

EXPRESS specification:

```
*)
RULE approval_requires_approval_date_time
   FOR (approval, approval_date_time);
WHERE
  wr1: SIZEOF (QUERY (app <* approval |
                  NOT (SIZEOF (QUERY (adt <* approval_date_time |
                      app :=: adt.dated_approval )) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

approval: identifies the set of all instances of **approval** entities.

approval_date_time: identifies the set of all instances of **approval_date_time** entities.

Formal propositions:

WR1: For each instance of **approval**, there shall be exactly one instance of **approval_date_time** which contains the instance of **approval** as its **dated_approval** attribute.

5.2.4.4 approval_requires_approval_person_organization

The **approval_requires_approval_person_organization** specifies that each instance of **approval** shall have at least one **approval_person_organization** referencing it except for approval instances that act as collectors of other approval instances. These collector approvals will be associated with the other approvals through approval_relationship entities. The collector approval will be related to the approval_relationship through the approval_relationship.relating_approval attribute. This rule enforces the requirement for an **approval** to be authorized by one or more people within their organizations.

NOTE This rule applies to mapping table rule 3.

EXPRESS specification:

```
*)
RULE approval_requires_approval_person_organization
   FOR (approval, approval_person_organization);
WHERE
  wr1: SIZEOF(QUERY(app <* approval | NOT (SIZEOF(QUERY(apo <*
   approval_person_organization |
   app :=: apo.authorized_approval )) >= 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

approval: identifies the set of all instances of **approval** entities.

approval_person_organization: identifies the set of all instances of **approval_person_organization** entities.

Formal propositions:

WR1: For each instance of **approval**, there shall be one or more instances of **approval_person_-organization** that contains the instance of **approval** as its **authorized_approval attribute**.

5.2.4.5 change_identification_restricts_executed_action

The **change_identification_restricts_executed_action** rule specifies that an **executed_action** with **executed_action.description** equal to 'change identification' shall have one or more **applied_action_assignment**s associated to itself.

NOTE This rule applies to mapping table rule 4.

EXPRESS specification:

```
*)
RULE change_identification_restricts_executed_action
     FOR (executed_action);
LOCAL
  result  : BOOLEAN := TRUE;
  found   : BOOLEAN := FALSE;
  exeact  : SET OF executed_action;
  acta    : BAG OF action_assignment;
END_LOCAL;
  exeact := executed_action;
  REPEAT i := 1 TO SIZEOF(exeact);
    IF (exeact[i].description = 'change identification') THEN
      acta := USEDIN(exeact[i],
     'TECHNICAL_DATA_PACKAGING.ACTION_ASSIGNMENT.ASSIGNED_ACTION');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(acta);
        IF (EXISTS(acta[j].role)) then
          IF ((acta[j].role.name = 'change identification') XOR
              (acta[j].role.name = 'revision history')) THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF (NOT found) THEN
        result := FALSE;
        ESCAPE;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

executed_action: identifies the set of all instances of **executed_action** entities.

Formal propositions:

WR1: An **executed_action** with **executed_action.description** equal to 'change identification' shall have one or more of the following associated to it:

— Is pointed to by a single **applied_action_assignment** with the **applied_action_assignment**'s associated object_role.name ='change identification'; OR

— Is pointed to by a single **applied_action_assignment** with the **applied_action_assignment**'s associated object_role.name ='revision history'.

5.2.4.6 compatible dimension

The **compatible_dimension** rule specifies that the count of coordinates of each **cartesian_point** matches the coordinate space dimension of each **geometric_representation_context** in which it is geometrically founded and that the count of firection ratios of each **direction** matches the coordinate space dimension of each **geometric_representation_context** in which it is geometrically founded.

NOTE This rule is from ISO 10303-42

EXPRESS Specification:

```
*)
RULE compatible_dimension FOR
  (cartesian_point,
  direction,
  representation_context,
  geometric_representation_context);
WHERE

  -- ensure that the count of coordinates of each cartesian_point
  -- matches the coordinate_space_dimension of each geometric_context in
  -- which it is geometrically founded
  WR1: SIZEOF(QUERY(x <* cartesian_point| SIZEOF(QUERY
       (y <* geometric_representation_context | item_in_context(x,y) AND
       (HIINDEX(x.coordinates) <> y.coordinate_space_dimension))) > 0 ))
         =0;

  -- ensure that the count of direction_ratios of each direction
  -- matches the coordinate_space_dimension of each geometric_context in
  -- which it is geometrically founded
  WR2: SIZEOF(QUERY(x <* direction | SIZEOF( QUERY
       (y <* geometric_representation_context | item_in_context(x,y) AND
       (HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension)))
       > 0 )) = 0;
END_RULE;
(*
```

Argument definitions:

cartesian_point: identifies the set of all instances of **cartesian_point** entities.

direction: identifies the set of all instances of **direction** entities.

representation_context: identifies the set of all instances of **representation_context** entities.

geometric_representation_context: identifies the set of all instances of **geometric_representation_-context** entities.

Formal propositions:

WR1: Ensures that the count of coordinates of each **cartesian_point** matches the coordinate space dimension of each **geometric_representation_context** in which it is geometrically founded

WR2: Ensures that the count of direction ratios of each **direction** matches the coordinate space dimension of each **geometric_representation_context** in which it is geometrically founded.

5.2.4.7 contract_submission_requires_date_and_organization

The **contract_submission_requires_date_and_organization** rule specifies how a **contract** being submitted must have date of submission and location of the destination organization.

NOTE 1 This rule applies to mapping table rule 5 and 6.

EXPRESS specification:

```
*)
RULE contract_submission_requires_date_and_organization
    FOR (applied_contract_assignment);
LOCAL
  result : BOOLEAN := TRUE;
  found : BOOLEAN;
  apc :  SET OF applied_contract_assignment;
  adta : BAG OF applied_date_and_time_assignment;
  ada :  BAG OF applied_date_assignment;
  aoa :  BAG OF applied_organization_assignment;
  oadr : BAG OF organizational_address;
  padr : BAG OF personal_address;
  apoa : BAG OF applied_person_and_organization_assignment;
END_LOCAL;
  apc := applied_contract_assignment;
  REPEAT i := 1 TO SIZEOF(apc);
    IF (EXISTS(apc[i]\contract_assignment.role)) THEN
     IF (apc[i]\contract_assignment.role.name = 'contract submission') THEN
        adta := USEDIN(apc[i], 'TECHNICAL_DATA_PACKAGING.'+
                       'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS');
        found := FALSE;
        REPEAT j := 1 TO SIZEOF(adta);
          IF (EXISTS(adta[j]\date_and_time_assignment.role)) then
            IF (adta[j]\date_and_time_assignment.role.name =
                'date and time of submission') THEN
              found := TRUE;
            END_IF;
          END_IF;
        END_REPEAT;
        IF (NOT found) THEN
          result := FALSE;
```

```
  ESCAPE;
END_IF;
ada := USEDIN(apc[i], 'TECHNICAL_DATA_PACKAGING.'+
              'APPLIED_DATE_ASSIGNMENT.ITEMS');
found := FALSE;
REPEAT j := 1 TO SIZEOF(ada);
  IF (EXISTS(ada[j]\date_assignment.role)) THEN
    IF (ada[j]\date_assignment.role.name = 'date of submission')
    THEN
      found := TRUE;
    END_IF;
  END_IF;
END_REPEAT;
IF (NOT found) THEN
  result := FALSE;
  ESCAPE;
END_IF;
aoa := USEDIN(apc[i], 'TECHNICAL_DATA_PACKAGING.'+
               'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS');
found := FALSE;
REPEAT j := 1 TO SIZEOF(aoa);
  IF (EXISTS(aoa[j]\organization_assignment.role)) then
    IF (aoa[j]\organization_assignment.role.name =
        'location of contract submission') THEN

        oadr := USEDIN (aoa[j]\
        organization_assignment.assigned_organization,
        'TECHNICAL_DATA_PACKAGING.'+
        'ORGANIZATIONAL_ADDRESS.ORGANIZATIONS');
       IF (SIZEOF(oadr)>=1)THEN
        found := TRUE;
       END_IF;
    END_IF;
  END_IF;
END_REPEAT;
IF (NOT found) THEN
  result := FALSE;
  ESCAPE;
END_IF;
apoa := USEDIN(apc[i], 'TECHNICAL_DATA_PACKAGING.'+
             'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS');
found := FALSE;
REPEAT j := 1 TO SIZEOF(apoa);
  IF (EXISTS(apoa[j]\person_and_organization_assignment.role)) then
    IF (apoa[j]\person_and_organization_assignment.role.name =
        'location of contract submission') THEN
        oadr := USEDIN (apoa[j]\person_and_organization_assignment.
           assigned_person_and_organization.the_organization,
           'TECHNICAL_DATA_PACKAGING.'+
           'ORGANIZATIONAL_ADDRESS.ORGANIZATIONS');
       IF (SIZEOF(oadr)>=1)THEN
        found := TRUE;
       END_IF;
       padr := USEDIN (apoa[j]\person_and_organization_assignment.
             assigned_person_and_organization.the_person,
        'TECHNICAL_DATA_PACKAGING.PERSONAL_ADDRESS.PEOPLE');
       IF (SIZEOF(padr)>=1)THEN
        found := TRUE;
       END_IF;
```



```
            END_IF;
          END_IF;
        END_REPEAT;
        IF (NOT found) THEN
          result := FALSE;
          ESCAPE;
        END_IF;

      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

applied_contract_assignment: identifies the set of all instances of **applied_contract_assignment** entities.

Formal propositions:

WR1: If an **applied_contract_assignment** that has its associated **object_role.name**='contract submission', then:

— the **applied_contract_assignment** shall be pointed to by an **applied_date_and_time_assignment** where the **data_time_role.name** ='date and time of submission'; OR

— the **applied_contract_assignment** shall be pointed to by an **applied_date_assignment** where the **data_-role.name** ='date of submission'.

NOTE 2 This rule applies to mapping table rule 5.

WR2: If an **applied_contract_assignment** that has its associated **object_role.name** ='contract submission', then the **applied_contract_assignment** shall be pointed to by an **applied_organization_-assignment** where the **organization_role.name** ='location of contract submission' and the **applied_-organization_assignment** is pointed to by an **organizational_address**. The **organizational_address** represents the location where the **contract** is submitted.

NOTE 3 This rule applies to mapping table rule 6.

5.2.4.8 data_definition_entry_string_restrict_for_superseded_element

The **data_definition_entry_string_restrict_for_superseded_element** rule specifies how a superseded relationship is identified through the restriction of a attribute or a **product_definition_formation_-relationship.name** atribute.

NOTE This rule applies to mapping table rule 7.

EXPRESS specification:

```
*)
RULE data_definition_entry_string_restrict_for_superseded_element
FOR (product_definition) ;
LOCAL
  found : BOOLEAN;
  result1 : BOOLEAN;
  result2 : BOOLEAN;
  pd : SET OF product_definition;
  pdca : BAG OF product_definition_context_association;
  pdc : product_definition_context;
  pdf : product_definition_formation;
END_LOCAL;
  result1 := TRUE;
  result2 := TRUE;
  pd :=product_definition;
  REPEAT i := 1 TO SIZEOF (pd);
   found :=FALSE;
   pdca :=USEDIN (pd[i], 'TECHNICAL_DATA_PACKAGING.'+
         'PRODUCT_DEFINITION_CONTEXT_ASSOCIATION.DEFINITION');
   REPEAT j :=1 TO SIZEOF (pdca);
      pdc :=pdca [j].frame_of_reference;
      IF (pdc\application_context_element.name IN
         ['document version',
          'a representation of a document version',
          'data definition entry',
          'indentured data list entry']) THEN
        found :=TRUE;
      END_IF;
  END_REPEAT;
  IF (found) THEN
    IF ((SIZEOF (QUERY(pdr <* USEDIN(pd[i], 'TECHNICAL_DATA_PACKAGING.'+
                  'PRODUCT_DEFINITION_RELATIONSHIP.'+
                  'RELATED_PRODUCT_DEFINITION') |
        (pdr.name = 'superseded element')))) > 1) THEN
   result1 :=FALSE;
    END_IF;
    pdf := pd[i].formation;
    IF ((SIZEOF(QUERY(pdfr <* USEDIN(pdf, 'TECHNICAL_DATA_PACKAGING.'+
                  'PRODUCT_DEFINITION_FORMATION_RELATIONSHIP.'+
                  'RELATED_PRODUCT_DEFINITION_FORMATION') |
        (pdfr.name = 'superseded version')))) > 1) THEN
   result2 := FALSE;
    END_IF;
    END_IF;
    END_REPEAT;
WHERE
    wr1 : result1;
    wr2 : result2;
END_RULE;
(*
```

Argument definitions:

product_definition: identifies the set of all instances of **product_definition** entities.

Formal propositions:

WR1: A **product_definition** that has a related **product_definiton_context_association** that has an associated **application_context_element.name** =’document version’ OR an **application_context_-element.name** =’a representation of a document version’ OR an **application_context_element.name** =’data definition entry’ or an **application_context_element.name** =‘indentured data list entry’ shall have zero or one **product_definition_relationships** that have a **product_definition_relationship.name** =’superseded element’.

WR2: A **product_definition** that has a related **product_definiton_context_association** that has an associated **application_context_element.name** =’document version’ OR an **application_context_-element.name** =’a representation of a document version’ OR an **application_context_element.name** =’data definition entry’ or ‘indentured data list entry’ where the **product_definition.formation** points to a **product_definition_formation** which shall have zero or one **product_definition_formation_relationships** that have a **product_definition_formation_relationship.name** =’superseded version’.

5.2.4.9 data_definition_exchange_categorized_as_document

The **data_definition_exchange_categorized_as_document** rule specifies how a **data_definition_exchange** must be categorized as a document.

NOTE This rule applies to mapping table rule 8.

EXPRESS specification:

```
*)
RULE data_definition_exchange_categorized_as_document
   FOR (product_related_product_category);
LOCAL
  result : BOOLEAN := TRUE;
  prpc   : SET OF product_related_product_category;
  prds   : SET OF product := [];
  pdf    : BAG OF product_definition_formation;
  pc     : SET OF product_category;
END_LOCAL;
  prpc := QUERY(v <* product_related_product_category |
                v.name = 'data definition exchange');
  REPEAT i := 1 TO SIZEOF(prpc);
    prds := prpc[i].products;
    REPEAT j := 1 TO SIZEOF(prds);
      pdf := USEDIN(prds[j], ‘TECHNICAL_DATA_PACKAGING.’+
             'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT');
      IF (SIZEOF(pdf) > 0) THEN
        pc := product_category_graph_members(prpc[i]);
        IF (SIZEOF(QUERY(v <* pc | v.name = 'document')) < 1) THEN
          result := FALSE;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
```

```
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_related_product_category: identifies the set of all instances of **product_related_product_-category** entities.

Formal propositions:

WR1: A **product_definition_formation** where the **product_definition_formation.of_product** points to a **product** where the **product** is pointed at by a **product_related_product_category.products** and the **product_related_product_category.name** ='data definition exchange', then a second instance of **product_-related_product_category** shall exist in the same product category structure, established by **product_-category_relationships**, with the attribute **product_related_product_category.name** ='document'.

5.2.4.10 data_definition_exchange_restrict_to_one_presentation

The **data_definition_exchange_restrict_to_one_presentation** rule specifies how to insure that a **data_-definition_exchange** document will never have more than one presentation, captured within this part of ISO 10303.

NOTE This rule applies to mapping table rule 9.

EXPRESS specification:

```
*)
RULE data_definition_exchange_restrict_to_one_presentation
   FOR (product_related_product_category);
LOCAL
  result : BOOLEAN := TRUE;
  prpc   : SET OF product_related_product_category;
  prds   : SET OF product := [];
  pdf    : BAG OF product_definition_formation;
END_LOCAL;
  prpc := product_related_product_category;
  REPEAT i := 1 TO SIZEOF(prpc);
    IF (prpc[i]\product_category.name = 'data definition exchange') THEN
      prds := prds + prpc[i].products;
    END_IF;
  END_REPEAT;
  REPEAT i := 1 TO SIZEOF(prds);
    pdf := USEDIN(prds[i], 'TECHNICAL_DATA_PACKAGING.'+
                  'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT');
    REPEAT j := 1 TO SIZEOF(pdf);
      IF (SIZEOF(USEDIN(pdf[j], 'TECHNICAL_DATA_PACKAGING.'+
                        'APPLIED_PRESENTED_ITEM.ITEMS')) > 1) THEN
        result := FALSE;
      END_IF;
    END_REPEAT;
  END_REPEAT;
WHERE
```



```
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_related_product_category: identifies the set of all instances of **product_related_product_-category** entities.

Formal propositions:

WR1: If a **product_definition_formation** has an associated **product_category.name** =’data definition exchange’, then the **product_definition_formation** may be pointed to by zero or one **presented_item**.

5.2.4.11 dependent_instantiable_named_unit

The **dependent_instantiable_named_unit** rule specifies that a **named_unit** may not be instantiated without being related to another entity.

NOTE 1 This rule applies to mapping table rule 10.

EXPRESS specification:

```
*)
RULE dependent_instantiable_named_unit
  FOR (named_unit);
LOCAL
  result : BOOLEAN := TRUE;
  nu     : SET OF named_unit;
END_LOCAL;
  nu := named_unit;
  REPEAT i := 1 TO SIZEOF(nu);
    IF (SIZEOF(USEDIN(nu[i], '')) < 1) THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

named_unit: identifies the set of all instances of **named_unit** entities.

Formal propositions:

WR1: For each instance of **named_unit**, there shall be a reference to the **named_unit** instance from an attribute of another entity.

NOTE 2 From ISO 10303-203 section 5.2.4.53 **dependent_instantiable_named_unit**.

5.2.4.12 distribution_notice_approval_requires_supporting_data

The **distribution_notice_approval_requires_supporting_data** rule specifies that an **approval** with an associated **applied_approval_assignment** with an **object_role.name** = 'distribution notice' shall have at least one piece of supporting data associated with itself.

NOTE This rule applies to mapping table rule 11.

EXPRESS specification:

```
*)
RULE distribution_notice_approval_requires_supporting_data
   FOR (applied_approval_assignment);
LOCAL
  result  : BOOLEAN := TRUE;
  found   : BOOLEAN;
  aaa     : SET OF applied_approval_assignment;
  ap      : approval;
  apo     : BAG OF approval_person_organization;
  aca     : BAG OF applied_classification_assignment;
END_LOCAL;
  aaa := applied_approval_assignment;
  REPEAT i := 1 TO SIZEOF(aaa);
    IF (EXISTS(aaa[i].role)) THEN
      IF (aaa[i].role.name = 'distribution notice') THEN
        ap := aaa[i].assigned_approval;
        found := FALSE;
        IF (LENGTH(ap.level) > 0) THEN
          found := TRUE;
        END_IF;
        apo := USEDIN(ap, 'TECHNICAL_DATA_PACKAGING.'+
                          'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL');
        REPEAT j := 1 TO SIZEOF(apo);
          IF (apo[j].role.role = 'distribution authorization') THEN
            found := TRUE;
          END_IF;
        END_REPEAT;
        aca := USEDIN(ap, 'TECHNICAL_DATA_PACKAGING.'+
                          'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS');
        REPEAT j := 1 TO SIZEOF(aca);
         IF (EXISTS(aca[j]\classification_assignment.role)) THEN
          IF ((aca[j]\classification_assignment.role.name =
                'distribution notice code') AND
               (('TECHNICAL_DATA_PACKAGING.CLASS') IN
TYPEOF(aca[j]\classification_assignment.assigned_class))) THEN
              found := TRUE;
            END_IF;
          END_IF;
        END_REPEAT;
        IF (NOT found) THEN
          result := FALSE;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
```



```
END_RULE;
(*
```

Argument definitions:

applied_approval_assignment: identifies the set of all instances of **applied_approval_assignment** entities.

Formal proposition:

WR1: Each instance of **approval** with an associated **applied_approval_assignment** with an **object_role.name** = 'distribution notice' shall have at least one of the following:

— an associated **class** with an **applied_classification_assignment** with a **classification_role.name** = 'distribution notice code';

— an **approval_person_organization** where **approval_person_organization.role** points to an **approval_role** where **approval_role.name** ='distribution authorization';

— an **approval.level** that is non-null.

5.2.4.13 document_product_equivalence_existence_rule

The **document_product_equivalence_existence_rule** rule specifies that the when a document references other product data through a **document_reference** or a **document_usage_constraint** a collection of entities need to be instantated. These entities include **product**, **product_definition_formation**, **document_product_equivalence**, **document**, **document_type**, and **product_related_product_category**.

NOTE This rule applies to mapping table rule 12.

EXPRESS specification:

```
RULE document_product_equivalence_existence_rule
  FOR (applied_document_reference,
       applied_document_usage_constraint_assignment);
LOCAL
  found  : BOOLEAN := FALSE;
  result : BOOLEAN := TRUE;
  adr    : SET OF applied_document_reference;
  aduca  : SET OF applied_document_usage_constraint_assignment;
  doc    : document;
  dpa    : BAG OF document_product_association;
  pfd    : product_or_formation_or_definition;
  dt     : label;
END_LOCAL;
  adr := applied_document_reference;
  REPEAT i := 1 TO SIZEOF(adr);
    doc := adr[i]\document_reference.assigned_document;
    dt := doc.kind.product_data_type;
    dpa := USEDIN(doc, 'TECHNICAL_DATA_PACKAGING.'+
          'DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT');
    REPEAT j := 1 TO SIZEOF(dpa);
      IF (('TECHNICAL_DATA_PACKAGING.DOCUMENT_PRODUCT_EQUIVALENCE')
          IN TYPEOF(dpa[j])) THEN
```

```
        found := TRUE;
        pfd := dpa[j].related_product;
        IF (('TECHNICAL_DATA_PACKAGING.PRODUCT') IN TYPEOF(pfd)) THEN
          IF (dt <> 'configuration controlled document') THEN
            result := FALSE;
          END_IF;
        ELSE
          IF (('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION')
              IN TYPEOF(pfd)) THEN
            IF (dt <> 'configuration controlled document version') THEN
              result := FALSE;
            END_IF;
          ELSE
            IF (('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION') IN
                  TYPEOF(pfd))
            THEN
              IF (dt <> 'configuration controlled document definition')
              THEN
                result := FALSE;
              END_IF;
            ELSE
              result := FALSE;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
    IF (NOT found) THEN
      (* There must be at least one *)
      result := FALSE;
    END_IF;
  END_REPEAT;
  aduca := applied_document_usage_constraint_assignment;
  REPEAT i := 1 TO SIZEOF(aduca);
    doc := aduca[i]\document_usage_constraint_assignment.
           assigned_document_usage.source;
    dt := doc.kind.product_data_type;
    dpa := USEDIN(doc, 'TECHNICAL_DATA_PACKAGING.'+
          'DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT');
    REPEAT j := 1 TO SIZEOF(dpa);
      IF (('TECHNICAL_DATA_PACKAGING.DOCUMENT_PRODUCT_EQUIVALENCE')
           IN TYPEOF(dpa[j])) THEN
        found := TRUE;
        pfd := dpa[j].related_product;
        IF (('TECHNICAL_DATA_PACKAGING.PRODUCT') IN TYPEOF(pfd)) THEN
          IF (dt <> 'configuration controlled document') THEN
            result := FALSE;
          END_IF;
        ELSE
          IF (('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION')
              IN TYPEOF(pfd))
          THEN
            IF (dt <> 'configuration controlled document version')
            THEN result := FALSE;
            END_IF;
          ELSE
            IF (('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION') IN
                 TYPEOF(pfd))
            THEN
```



```
            IF (dt <> 'configuration controlled document definition')
            THEN result := FALSE;
            END_IF;
          ELSE
            result := FALSE;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    END_REPEAT;
    IF (NOT found) THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
```

Argument definitions:

applied_document_reference: identifies the set of all instances of **applied_document_reference** entities.

Formal propositions:

WR1: If an **applied_document_reference** or an **applied_document_usage_constraint** exist, then the associated **document** instance shall be associated to a **product**, **product_definition_formation**, or a **product_definition** through the entity **document_product_equivalence**. The **document_product_-equivalence** instance will have its name attribute equal to 'equivalence'. The **document** instance shall have an associated **document_type.product_data_type** to either 'configuration controlled document', 'configuration controlled document version', or 'configuration controlled document definition' based on what **document_product_equivalence** is pointing at, either a **product**, **product_definition_formation**, or **product_definition**, respectively.

5.2.4.14 drawing_suffix_number_combination_identification_constraint

The **drawing_suffix_number_combination_identification_constraint** rule specifies how to identify the document in a drawing suffix number combination relationship.

NOTE This rule applies to mapping table rule 13.

EXPRESS specification:

```
*)
RULE drawing_suffix_number_combination_identification_constraint
    FOR (product_relationship);
LOCAL
  result  : BOOLEAN := TRUE;
  tresult : BOOLEAN;
  pr      : SET OF product_relationship;
  p       : product;
  prpc    : BAG OF product_related_product_category;
END_LOCAL;
  pr := QUERY(v <* product_relationship |
```

```
              v.name = 'drawing suffix number combination');
  REPEAT i := 1 to SIZEOF(pr);
    p := pr[i].relating_product;
    prpc := USEDIN(p, 'TECHNICAL_DATA_PACKAGING.'+
                    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    REPEAT j := 1 TO SIZEOF(prpc);
      IF (prpc[j]\product_category.name = 'document') THEN
        result := FALSE;
      END_IF;
    END_REPEAT;
    p := pr[i].related_product;
    prpc := USEDIN(p, 'TECHNICAL_DATA_PACKAGING.'+
                    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    tresult := FALSE;
    REPEAT j := 1 TO SIZEOF(prpc);
      IF (prpc[j]\product_category.name = 'document') THEN
        tresult := TRUE;
      END_IF;
    END_REPEAT;
    IF (NOT tresult) THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_relationship: identifies the set of all instances of **product_relationship** entities.

Formal propositions:

WR1: This rule applies if a part identification has the name of the drawing that defines it as a portion of the part's identification and this fact needs to be captured. If a **product_relationship** exist with **product_-relationshipname** equal to 'drawing suffix number combination' then its **product_relationship.relating_-product** will point to a **product** without an associated **product_related_product_category.name** ='document' and its **product_relationship.related_product** will point to a **product** with an associated **product_related_product_category.name** ='document'.

5.2.4.15 exchange_reason_existence_required

The **exchange_reason_existence_required** rule specifies that an exchange reason needs to exist for a **data_definition_exchange** document.

NOTE This rule applies to mapping table rule 14.

EXPRESS specification:

```
*)
RULE exchange_reason_existence_required
    FOR (property_definition);
LOCAL
```



```
  result : BOOLEAN  := TRUE;
  prpd   : SET OF property_definition;
  prdr   : BAG OF property_definition_representation;
  repi   : representation;
  dri    : descriptive_representation_item;
END_LOCAL;
  prpd := property_definition;
  REPEAT i := 1 TO SIZEOF(prpd);
    IF (prpd[i].name = 'data definition exchange header') THEN
      result := FALSE;
      prdr := USEDIN(prpd[i],'TECHNICAL_DATA_PACKAGING.'+
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
      REPEAT j := 1 TO SIZEOF (prdr);
        repi := prdr[j].used_representation;
        IF (NOT (repi.name = 'exchange reason')) THEN
          ESCAPE;
        ELSE -- (cycle through representation items)
          REPEAT k := 1 TO SIZEOF (repi.items);
            IF ('TECHNICAL_DATA_PACKAGING.DESCRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(repi.items[k])) THEN
              dri := repi.items[k];
              IF ((dri.name = 'exchange purpose') XOR
                  (dri.name = 'base exchange reason')) THEN
                     -- (having an additional dri.name = 'exchange purpose'
                     -- is irrelevant since the above condition already
                     -- entail a TRUE condition)
                result := TRUE;
                ESCAPE;
              END_IF;
            END_IF;
          END_REPEAT;
        END_IF;
      END_REPEAT;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

property_definition: identifies the set of all instances of **property_definition** entities.

Formal propositions:

WR1: A **property_definition** with **property_definition.name** ='data definition exchange header' shall have an associated **representation** with **representation.name** ='exchange reason'. This **representation** shall have either one associated descriptive_representation_item with **descriptive_representation_item.name** ='base exchange reason' OR one associated **descriptive_representation_item** with **descriptive_representation_item.name** ='exchange purpose', OR two associated **descriptive_representation_item**, one with **descriptive_representation_item.name** ='base exchange reason' and the other with **descriptive_representation_item.name** ='exchange purpose'.

5.2.4.16 existence_dependent_drawing_sheet_revision

The **existence_dependent_drawing_sheet_revision** rule specifies that a **drawing_sheet_revision** shall alway be associated with a **product** that is of **product_category** named ‘sheet’.

NOTE This rule applies to mapping table rule 15.

EXPRESS specification:

```
*)
RULE existence_dependent_drawing_sheet_revision
    FOR (drawing_sheet_revision);
LOCAL
  result : BOOLEAN := TRUE;
  dsr    : SET OF drawing_sheet_revision;
  pir    : BAG OF presented_item_representation;
  pdf    : SET OF product_definition_formation;
  prpc   : BAG OF product_related_product_category;
END_LOCAL;
  dsr := drawing_sheet_revision;
  REPEAT i := 1 TO SIZEOF(dsr);
    pir := USEDIN(dsr[i], ‘TECHNICAL_DATA_PACKAGING.’+
                  'PRESENTED_ITEM_REPRESENTATION.PRESENTATION');
      IF (SIZEOF(pir) < 1) THEN
        result := FALSE;
      END_IF;
      REPEAT j := 1 TO SIZEOF(pir);
        pdf := pir[j].item\applied_presented_item.items;
        IF (SIZEOF(pdf) < 1) THEN
          result := FALSE;
        END_IF;
        REPEAT k := 1 TO SIZEOF(pdf);
          prpc := USEDIN(pdf[k].of_product, ‘TECHNICAL_DATA_PACKAGING.’ +
                         'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
          IF (SIZEOF(prpc) < 1) THEN
            result := FALSE;
          END_IF;
          REPEAT l := 1 TO SIZEOF(prpc);
            IF (prpc[l]\product_category.name <> 'sheet') THEN
              result := FALSE;
            END_IF;
          END_REPEAT;
        END_REPEAT;
      END_REPEAT;
    END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

drawing_sheet_revision: identifies the set of all instances of **drawing_sheet_revision** entities.

Formal propositions:

WR1: If a **drawing_sheet_revision** exist, then it will be pointed at by a **presented_item_representation.presentation**. Also the **presented_item_representation** will point to an **applied_presented_item** that in turn will point at a **product_definition_formation** with an associated **product_related_product_category.name** =’sheet’.

5.2.4.17 file_format_restricted_to_one_data_definition_entry

The **file_format_restricted_to_one_data_definition_entry** rule specifies how only one file format can be associated directly with a Data_definition_entry_tdp_element.

NOTE This rule applies to mapping table rule 16.

EXPRESS specification:

```
*)
RULE file_format_restricted_to_one_data_definition_entry
    FOR (product_definition);
LOCAL
  result   :  BOOLEAN  := TRUE;
  pd       :  SET OF product_definition;
  propd : BAG OF property_definition;
END_LOCAL;
  pd := product_definition;
  REPEAT i := 1 TO SIZEOF(pd);
    IF ((pd[i].frame_of_reference.name = 'document version') XOR
        (pd[i].frame_of_reference.name =
                'a representation of a document version')) THEN
      propd := USEDIN(pd[i], ‘TECHNICAL_DATA_PACKAGING.'+
                      'PROPERTY_DEFINITION.DEFINITION');
      IF (SIZEOF(QUERY(p <* propd | p.name = 'document format')) > 1) THEN
        result := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_definition: identifies the set of all instances of **product_definition** entities.

Formal propositions:

WR1: A **product_definition** or **product_definition_with_associated_document** that has a **product_definition_context_association** that has an **application_context_element.name** =‘document version’ or an **application_context_element.name** =‘a representation of a document version’ shall have zero or one **property_definition** where the **property_definition.name** =‘document format’.

5.2.4.18 header_configuration_restricts_property_definition

The **header_configuration_restricts_property_definition** rule specifies that a **property_definition** with **property_definition.name** equal to 'xxx header' (with the word 'header' being the last set of characters in the string) shall have one or more of the following associated to it:

— **applied_approval_assignment**;

— **property_definition_representation**.

NOTE This rule applies to mapping table rule 17.

EXPRESS specification:

```
*)
RULE header_configuration_restricts_property_definition
    FOR (property_definition);
LOCAL
  result  : BOOLEAN := TRUE;
  found   : BOOLEAN;
  pd      : SET OF property_definition;
  apa     : BAG OF applied_approval_assignment;
END_LOCAL;
  pd := QUERY(v <* property_definition | v.name LIKE '* header');
  REPEAT i := 1 TO SIZEOF(pd);
    found := FALSE;
    apa := USEDIN(pd[i],
   'TECHNICAL_DATA_PACKAGING.APPLIED_APPROVAL_ASSIGNMENT.ITEMS');
    REPEAT j := 1 TO SIZEOF(apa);
      IF (EXISTS(apa[j]\approval_assignment.role)) THEN
        IF ((apa[j]\approval_assignment.role.name = 'release
authentication') OR
            (apa[j]\approval_assignment.role.name = 'data usage rights'))
THEN
          found := TRUE;
        END_IF;
      END_IF;
    END_REPEAT;
    IF (SIZEOF(USEDIN(pd[i], 'TECHNICAL_DATA_PACKAGING.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) > 0)
THEN
      found := TRUE;
    END_IF;
    IF (NOT found) THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

property_definition: identifies the set of all instances of **property_definition** entities.

Formal proposition:

WR1: A **property_definition** with **property_definition.name** equal to 'xxx header' (with the 'header' being the last set of characters in the string) shall have one or more of the following associated to it:

— an **applied_approval_assignment** where **object_role.name** ='release authentication';

— an **applied_approval_assignment** where **object_role.name** ='data usage rights'; OR

— a **property_definition_representation**.

5.2.4.19 identification_of_sheet_constraint

The **identification_of_sheet_constraint** rule specifies the string value that identifies a sheet of a document that is mapped as a **product**.

NOTE This rule applies to mapping table rule 18.

EXPRESS specification:

```
*)
RULE identification_of_sheet_constraint
    FOR (product_definition_formation);
LOCAL
  found  : BOOLEAN;
  result : BOOLEAN := TRUE;
  pdf    : SET OF product_definition_formation;
  prpc   : BAG OF product_related_product_category;
  pc     : SET OF product_category;
END_LOCAL;
  pdf := product_definition_formation;
  REPEAT i := 1 TO SIZEOF(pdf);
    prpc := USEDIN(pdf[i].of_product, 'TECHNICAL_DATA_PACKAGING.' +
                    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    REPEAT j := 1 TO SIZEOF(prpc);
      IF (prpc[j]\product_category.name = 'sheet') THEN
        pc := product_category_graph_members(prpc[j]);
        found := FALSE;
        REPEAT k := 1 TO SIZEOF(pc);
          IF (pc[k].name = 'document') THEN
            found := TRUE;
          END_IF;
        END_REPEAT;
        IF (NOT found) THEN
          result := FALSE;
          ESCAPE;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```


Argument definitions:

product_definition_formation: identifies the set of all instances of **product_definition_formation** entities.

Formal propositions:

WR1: A **product_definition_formation** where the **product_definition_formation.of_product** points to a **product** where the **product** is pointed at by a **product_related_product_category.products** and the **product_category.name** ='sheet', then a second instance of **product_category** shall exist in the same product category structure, established by **product_category_relationships**, with the attribute **product_-category.name** ='document'. **Product_related_product_category**, a subtype of **product_category**, can be used instead of an instance of **product_category** in this rule.

5.2.4.20 indentured_data_list_identification_constraint

The **indentured_data_list_identification_constraint** rule specifies the string values in **product_category** that identify Indentured_data_lists.

NOTE This rule applies to mapping table rule 19.

EXPRESS specification:

```
*)
RULE indentured_data_list_identification_constraint
  FOR (product_definition_formation);
LOCAL
  result : BOOLEAN := TRUE;
  pdf : SET OF product_definition_formation;
  prpc : BAG OF product_related_product_category;
  pc : SET OF product_category;
END_LOCAL;
  pdf := product_definition_formation;
  REPEAT i := 1 TO SIZEOF(pdf);
    prpc := USEDIN(pdf[i].of_product,
   'TECHNICAL_DATA_PACKAGING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    REPEAT j  := 1 TO SIZEOF(prpc);
      IF (prpc[j]\product_category.name = 'indentured data list') THEN
        pc := product_category_graph_members(prpc[j]);
        result := FALSE;
        REPEAT k := 1 TO SIZEOF(pc);
          IF (pc[k].name = 'document') THEN
            result := TRUE;
          END_IF;
        END_REPEAT;
        IF (result = FALSE) THEN
          ESCAPE;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```


Argument definitions:

product_definition_formation: identifies the set of all instances of **product_definition_formation** entities.

Formal propositions:

WR1: A **product_definition_formation** where the **product_definition_formation.of_product** points to a **product** where the **product** is pointed at by a **product_related_product_category.products** and the **product_category.name** =’indentured data list’, then a second instance of **product_related_product_-category** shall exist in the same product category structure, established by **product_category_-relationship**s, with the attribute **product_related_product_category.name**= ‘document’.

5.2.4.21 indentured_level_tag_identification_constraint

The **indentured_level_tag_identification_constraint** rule specifies the string value used to identify a human interpretable indentured level for a Data_definition_entry and Indentured_data_list_entry.

NOTE This rule applies to mapping table rule 20.

EXPRESS specification:

```
*)
RULE indentured_level_tag_identification_constraint
    FOR (property_definition_representation);
LOCAL
  result : BOOLEAN := TRUE;
  pdr    : SET OF property_definition_representation;
END_LOCAL;
  pdr := property_definition_representation;
  REPEAT  i := 1 TO SIZEOF(pdr);
    IF (pdr[i].used_representation.name = 'indentured level tag') THEN
      IF NOT ((pdr[i].definition.name = 'exchange entry property') XOR
              (pdr[i].definition.name =
                      'indentured data list entry property')) THEN
        result := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

property_definition_representation: identifies the set of all instances of **property_definition_-representation** entities.

Formal propositions:

WR1: For any **representation.name** =’indentured level tag’, the associated **property_definition.name** =’exchange entry property’ or ‘indentured data list entry property’

5.2.4.22 indentured_list_method_identification_constraint

The **indentured_list_method_identification_constraint** rule specifies that the three different types of indentured list methods may be applied to only **data_definition_exchanges** and **indentured_data_lists**.

NOTE This rule applies to mapping table rule 21.

EXPRESS specification:

```
*)
RULE indentured_list_method_identification_constraint
  FOR (applied_document_reference);
LOCAL
  result : BOOLEAN := TRUE;
  adr : SET OF applied_document_reference;
  d : document;
  dpa : BAG OF document_product_association;
  pfd : product_or_formation_or_definition;
  p : product;
  prpc : BAG OF product_related_product_category;
END_LOCAL;
  adr := applied_document_reference;
  REPEAT i := 1 TO SIZEOF(adr);
   IF (EXISTS(adr[i]\document_reference.role)) THEN
    IF ((adr[i]\document_reference.role.name = 'indentured by document') OR
       (adr[i]\document_reference.role.name = 'indentured by item') OR
       (adr[i]\document_reference.role.name =
                    'indentured by item and document'))
THEN
        d := adr[i]\document_reference.assigned_document;
        dpa := USEDIN(d, ‘TECHNICAL_DATA_PACKAGING.'+
                      'DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT');
        REPEAT j := 1 TO SIZEOF(dpa);
          pfd := dpa[j].related_product;
          IF (‘TECHNICAL_DATA_PACKAGING.PRODUCT' IN TYPEOF(pfd)) THEN
            p := pfd;
          ELSE
           IF (‘TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION' IN
TYPEOF(pfd)) THEN
              p := pfd\product_definition_formation.of_product;
            ELSE
              IF (‘TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION' IN
TYPEOF(pfd)) THEN
              p := pfd\product_definition.formation.of_product;
              END_IF;
            END_IF;
          END_IF;
          result := FALSE;
          prpc := USEDIN(p, ‘TECHNICAL_DATA_PACKAGING.'+
```



```
          'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
          REPEAT k := 1 TO SIZEOF(prpc);
            IF ((prpc[k].name = 'data definition exchange') OR
                (prpc[k].name = 'indentured data list')) THEN
              result := TRUE;
            END_IF;
          END_REPEAT;
        END_REPEAT;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

applied_document_reference: identifies the set of all instances of **applied_document_reference** entities.

Formal propositions:

WR1: For an **applied_document_reference**, if its associated **object_role.name** ='indentured by document' or 'indentured by item' or 'indentured by item and document', then the associated **document** with its equivalent **product_definition** will have a corresponding **product_related_product_category.name** ='data definition exchange' or 'indentured data list'.

5.2.4.23 item_source_information_identification_constraint

The **item_source_information_identification_constraint** rule specifies the string values that identify the item source information.

NOTE This rule applies to mapping table rule 22.

EXPRESS specification:

```
*)
RULE item_source_information_identification_constraint
  FOR (property_definition);
LOCAL
  result : BOOLEAN  := TRUE;
  prpd   : SET OF property_definition;
  prdr   : BAG OF property_definition_representation;
END_LOCAL;
  prpd := property_definition;
  REPEAT i := 1 TO SIZEOF(prpd);
    IF (prpd[i].name = 'source information') THEN
      result := FALSE;
      prdr := USEDIN(prpd[i],'TECHNICAL_DATA_PACKAGING.'+
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
      REPEAT j := 1 TO SIZEOF(prdr);
        IF (prdr[j].used_representation.name = 'source information type')
THEN
          result := TRUE;
        END_IF;
```



```
      END_REPEAT;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

property_definition: identifies the set of all instances of **property_definition** entities.

Formal propositions:

WR1: A **property_definition** with the **property_definition.name** =’source information’ shall have an associated **representation** with the **representation.name** =’source information type’.

5.2.4.24 notation_type_identification_constraint

The **notation_type_identification_constraint** rule specifies the string values required to identify **notation** and their types.

NOTE This rule applies to mapping table rule 23.

EXPRESS specification:

```
*)
RULE notation_type_identification_constraint
  FOR (representation);
LOCAL
  result : BOOLEAN;
  note : SET OF representation;
END_LOCAL;
  note := QUERY(rep <* representation | rep.name='notation');
  result := FALSE;
   IF (SIZEOF (note)< 1)THEN
     result := TRUE;
   END_IF;
  REPEAT i := 1 TO SIZEOF(note);
    REPEAT j := 1 TO SIZEOF(note[i].items);
      IF (‘TECHNICAL_DATA_PACKAGING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(note[i].items[j])) then
        result := TRUE;
        ESCAPE;
      END_IF;
    END_REPEAT;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```


Argument definitions:

representation: identifies the set of all instances of **representation** entities.

Formal propositions:

WR1: Each **representation** with the **representation.name** ='notation' shall have a one or more associated **descriptive_representation_item**.

5.2.4.25 product_requires_category

The **product_requires_category** rule specifies that each instance of **product** shall be referenced by at least one **product_related_product_category**.

NOTE This rule applies to mapping table rule 24.

EXPRESS specification:

```
*)
RULE product_requires_category
   FOR (product,
        product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY (p <* product | SIZEOF (USEDIN (p,
      'TECHNICAL_DATA_PACKAGING.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS'
      ) ) =0 ) ) =0;
END_RULE;
(*
```

Argument definitions:

product: the set of all instances of **product**.

product_related_product_category : the set of all instances of **product_related_product_category**.

Formal propositions:

WR1: Each instance of **product** shall be referenced by at least one **product_related_product_category**.

5.2.4.26 product_requires_version

This **product_requires_version** rule specifies that each instance of the **product** shall be referenced by at least one **product_definition_formation**.

EXPRESS Specification:

```
*)
 RULE product_requires_version FOR (product);
  WHERE
   wr1 : SIZEOF(QUERY(prod <* product |
         (SIZEOF(USEDIN(prod,'TECHNICAL_DATA_PACKAGING.' +
```



```
          'PRODUCT_DEFINITION_FORMATION.' + 'OF_PRODUCT')) = 0))) = 0;
 END_RULE;
(*
```

Argument definitions:

product: the set of all instances of **product**.

Formal propositions:

WR1: Each instance of **product** shall be referenced by at least one **product_definition_formation**.

5.2.4.27 product_version_requires_person_organization

The **product_version_requires_person_organization** rule specifies that a **product_definition_formation** requires the association with an **organization** or **person_and_organization**. The organization and person is a parameter that aids in uniquely identifying the **product_definition_formation**.

NOTE 1 More than one organization or person_and_organization may be associated with a product_definition_-formation. This rule specifies that at least one of those organization_assignment.role or person_and_-organization.role attrubutes have the value of ‘creator’ or ‘id owner’.

NOTE 2 This rule applies to mapping table rule 25.

EXPRESS specification:

```
*)
RULE product_version_requires_person_organization
   FOR (product_definition_formation);
LOCAL
  found  : BOOLEAN;
  result : BOOLEAN := TRUE;
  pdf    : SET OF product_definition_formation;
  apos   : BAG OF applied_person_and_organization_assignment;
  apoa   : BAG OF applied_organization_assignment;
END_LOCAL;
  pdf := product_definition_formation;
  REPEAT i := 1 TO SIZEOF(pdf);
    apos := USEDIN(pdf[i], ‘TECHNICAL_DATA_PACKAGING.’ +
                    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS');
    found := FALSE;
    REPEAT j := 1 TO SIZEOF(apos) BY 1;
      IF (EXISTS(apos[j].role)) THEN
        IF ((apos[j].role.name = 'creator') OR
            (apos[j].role.name = 'id owner')) THEN
          found := TRUE;
        END_IF;
      END_IF;
    END_REPEAT;
    IF (found) THEN
      result := TRUE;
      ESCAPE;
    END_IF;
    apoa := USEDIN(pdf[i], ‘TECHNICAL_DATA_PACKAGING.’ +
```



```
                'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS');
    REPEAT j := 1 TO SIZEOF(apoa) BY 1;
      IF (EXISTS(apoa[j].role)) THEN
        IF ((apoa[j].role.name = 'creator') OR
            (apoa[j].role.name = 'id owner')) THEN
          found := TRUE;
        END_IF;
      END_IF;
    END_REPEAT;
    IF (NOT found) THEN
      result := FALSE;
      ESCAPE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_definition_formation: identifies the set of all instances of **product_definition_formation** entities.

Formal propositions:

WR1: A **product_definition_formation** shall be pointed to by one of the following:

— a **person_and_organization_assignment** where the **person_and_organization_assignment.role** points to a **person_and_organization_role** and the **person_and_organization_role.name** = ‘creator’ or ‘id owner’;

— an **organization_assignment** where **organization_assignment.role** points to an **organization_role** where the **organization_role.name** =‘creator’ or ‘id owner’.

5.2.4.28 reference_document_requires_subcategorization

The **reference_document_requires_subcategorization** rule specifies that a **product_related_product_-category.name=’reference document’** requires a lower level categorization using **product_category**.

NOTE This rule applies to mapping table rule 26.

EXPRESS specification:

```
*)
RULE reference_document_requires_subcategorization
    FOR (product_related_product_category);
LOCAL
  result : BOOLEAN := TRUE;
  prpc : SET OF product_related_product_category;
  pcr : BAG OF product_category_relationship;
END_LOCAL;
  prpc := QUERY(v <* product_related_product_category |
                          v\product_category.name = 'reference document');
  REPEAT i := 1 TO SIZEOF(prpc);
```

```
  pcr := USEDIN(prpc[i], 'TECHNICAL_DATA_PACKAGING.'+
                  'PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY');
    IF SIZEOF(pcr) < 1 THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

product_related_product_category: identifies the set of all instances of **product_related_product_-category** entities.

Formal propositions:

WR1: A product_related_product_category.name='reference document' shall never be instanced without an associated subcategory.

5.2.4.29 release_authentication_string_restriction

The **release_authentication_string_restriction** rule specifies how certain attribute string values in **approval**, **approval_role**, **identification_role**, and **approval_date_time_role** denote specific release authentication parameters.

NOTE 1 This rule applies to mapping table rule 27, 28, and 29.

EXPRESS specification:

```
*)
RULE release_authentication_string_restriction
  FOR (approval);
LOCAL
  result1 : BOOLEAN := TRUE;
  result2 : BOOLEAN := TRUE;
  found   : BOOLEAN;
  ar      : STRING := 'release authentication';
  appr    : SET OF approval;
  aaa     : BAG OF applied_approval_assignment;
  apo     : BAG OF approval_person_organization;
  adt     : BAG OF approval_date_time;
END_LOCAL;
  appr := approval;
  (* For each approval *)
  REPEAT i := 1 TO SIZEOF(appr);
    aaa := USEDIN(appr[i],'TECHNICAL_DATA_PACKAGING.'+
                  'APPROVAL_ASSIGNMENT.ASSIGNED_APPROVAL');
    (* Look for a release authentication approval assignment *)
    found := FALSE;
    REPEAT j := 1 TO SIZEOF(aaa);
      IF (EXISTS(aaa[j]\approval_assignment.role)) THEN
        IF (aaa[j].role.name = ar) THEN
```



```
          found := TRUE;
        END_IF;
      END_IF;
    END_REPEAT;
    IF (found) THEN
      (* Check for corresponding approval person organization *)
      apo := USEDIN(appr[i], 'TECHNICAL_DATA_PACKAGING.'+
                    'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(apo);
        IF (EXISTS(apo[j].role)) THEN
          IF (apo[j].role.role = ar) THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF (NOT found) THEN
        result1 := FALSE;
      END_IF;
      (* Check for corresponding approval date time *)
      adt := USEDIN(appr[i], 'TECHNICAL_DATA_PACKAGING.'+
                    'APPROVAL_DATE_TIME.DATED_APPROVAL');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(adt);
        IF (EXISTS(adt[j].role)) then
          IF (adt[j].role.name = ar) THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF (NOT found) THEN
        result2 := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result1;
  wr2 : result2;
END_RULE;
(*
```

Argument definitions:

applied_approval_assignment: identifies the set of all instances of **applied_approval_assignment** entities.

Formal propositions:

WR1: An **approval,** where its **applied_approval_assignment**'s associated **object_role.name** ='release authentication', shall have a **organization** or a **person_and_organization** that points to it and the associated **approval_role.role** ='release authentication'.

NOTE 2 This rule applies to mapping table rule 27.

WR2: An **approval**, where its **applied_approval_assignment**'s associated **object_role.name** = 'release authentication' shall be pointed to by an **approval_date_time** where the associated **approval_date_time_-role.name** = 'release authentication'.

NOTE 3 This rule applies to mapping table rule 29.

Informal propositions:

IP1: An **approval**, where its **applied_approval_assignment**'s associated **object_role.name** ='release authentication' shall have an **organization** or a **person_and_organization** that points to it. The **organization** shall have zero or one **applied_identification_assignment** pointing at it with an associated **identificaiton_role.name** equal to 'alias'.

NOTE 4 This rule applies to mapping table rule 28.

5.2.4.30 security_classification_date_string_restriction

The **security_classification_date_string_restriction** rule specifies how certain attribute string values in **date_role** and **date_time_role** denote specific classification and declassification dates.

NOTE 1 This rule applies to mapping table rule 30 and 31.

EXPRESS specification:

```
*)
RULE security_classification_date_string_restriction
    FOR (applied_security_classification_assignment);
LOCAL
  result  : BOOLEAN  := TRUE;   -- If there is no "security_assignment",
                                -- constraint does not apply
  tresult : BOOLEAN;   -- Temporary result for each iteration
  asca    : SET OF applied_security_classification_assignment;
  adta    : BAG OF applied_date_and_time_assignment;
  ada     : BAG OF applied_date_assignment;
END_LOCAL;
  asca := applied_security_classification_assignment;
  REPEAT i := 1 TO SIZEOF (asca);
    adta := USEDIN(asca[i],'TECHNICAL_DATA_PACKAGING.'+
                    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS');
    tresult := FALSE;
    REPEAT j := 1 TO SIZEOF (adta);
      REPEAT k := 1 TO SIZEOF (adta[j].items);
        IF (adta[j].items[k] = asca[i]) THEN
          IF (EXISTS(adta[j]\date_and_time_assignment.role)) then
            IF (adta[j]\date_and_time_assignment.role.name =
                  'classification date') THEN
              tresult := TRUE;
            END_IF;
          END_IF;
          result := result AND tresult;
        END_IF;
      END_REPEAT;
    END_REPEAT;
    ada := USEDIN(asca[i],'TECHNICAL_DATA_PACKAGING.'+
                   'APPLIED_DATE_ASSIGNMENT.ITEMS');
    REPEAT j := 1 TO SIZEOF (ada);
      REPEAT k := 1 TO SIZEOF (ada[j].items);
        IF (ada[j].items[k] = asca[i]) THEN
          IF (EXISTS(ada[j]\date_assignment.role)) THEN
```



```
            IF (ada[j]\date_assignment.role.name = 'classification date')
THEN
              tresult := TRUE;
            END_IF;
          END_IF;
          result := result AND tresult;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
WHERE
  wr1: result;
END_RULE;
(*
```

Argument definitions:

applied_security_classification_assignment: identifies the set of all instances of **applied_security_-classification_assignment** entities.

Formal propositions:

WR1: An **applied_security_classification_assignment** shall be pointed to either:

— by a **date_assignment** where the **date_role.name** =’classification date’; OR

— by a **date_and_time_assignment** where the **date_time_role.name** = ‘classification date’.

NOTE 2 This rule applies to mapping table rule 30.

Informal proposition:

IP1: An **applied_security_classification_assignment** shall be pointed to by zero or one of either:

— by a **date_assignment** where the **date_role.name** =’declassification date’; OR

— by a **date_and_time_assignment** where the **date_time_role.name** = ‘declassification date’.

NOTE 3 This rule applies to mapping table rule 31.

5.2.4.31 simple_list_of_elements_constraint

The **simple_list_of_elements_constraint** rule specifies the valid entities with in the select type **document_-reference_item** when used to identify a simple list of elements for a **data_definition_exchange** list.

NOTE This rule applies to mapping table rule 32.

EXPRESS specification:

```
*)
RULE simple_list_of_elements_constraint
```



```
  FOR (applied_document_reference);
LOCAL
  result : BOOLEAN := TRUE;
  adr    : SET OF applied_document_reference;
  dri    : SET OF document_reference_item;
END_LOCAL;
  adr := applied_document_reference;
  REPEAT i := 1 TO SIZEOF(adr);
    IF (EXISTS(adr[i]\document_reference.role)) THEN
      IF (adr[i]\document_reference.role.name = 'list of elements') then
        dri := adr[i].items;
        IF (SIZEOF(dri) < 1) THEN
          result := FALSE;
        END_IF;
        REPEAT j := 1 TO SIZEOF(dri);
          IF (NOT
      ('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION' IN
TYPEOF(dri[j]))) THEN
            result := FALSE;
          END_IF;
        END_REPEAT;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

applied_document_reference: identifies the set of all instances of **applied_document_reference** entities.

Formal propositions:

WR1: For any **applied_document_reference** that has an associated **object_role.name** =’list of elements’, the set of **document_reference_items** pointed to by the **applied_document_reference** shall be constrained to be of type **product_definition_formation**.

5.2.4.32 simple_list_of_files_constraint

The **simple_list_of_files_constraint** rule specifies the valid entities with in the select type **document_-reference_item** when used to identify a simple list of files for a Data_definition_exchange list.

NOTE This rule applies to mapping table rule 33.

EXPRESS specification:

```
*)
RULE simple_list_of_files_constraint
    FOR (applied_document_reference);
LOCAL
  result : BOOLEAN := TRUE;
  nameok : BOOLEAN;
  adr    : SET OF applied_document_reference;
```



```
  dri    : SET OF document_reference_item;
  drt    : SET OF document_representation_type;
END_LOCAL;
  adr := applied_document_reference;
  REPEAT i := 1 TO SIZEOF(adr);
    IF (EXISTS(adr[i]\document_reference.role)) THEN
      IF (adr[i]\document_reference.role.name = 'list of files') THEN
        dri := adr[i].items;
        IF (SIZEOF(dri) < 1) THEN
          result := FALSE;
        END_IF;
        REPEAT j := 1 TO SIZEOF(dri);
          IF (NOT ('TECHNICAL_DATA_PACKAGING.DOCUMENT_FILE' IN
TYPEOF(dri[j]))) THEN
            result := FALSE;
          END_IF;
          drt := dri[j].representation_types;
          nameok := FALSE;
          REPEAT k := 1 TO SIZEOF(drt);
            IF ((drt[k].name = 'digital') OR (drt[k].name = 'physical'))
         THEN
              nameok := TRUE;
            END_IF;
          END_REPEAT;
          IF (NOT nameok) THEN
            result := FALSE;
          END_IF;
        END_REPEAT;
      END_IF;
    END_IF;
  END_REPEAT;
WHERE
  wr1 : result;
END_RULE;
(*
```

Argument definitions:

applied_document_reference: identifies the set of all instances of **applied_document_reference** entities.

Formal propositions:

WR1: For any **applied_document_reference** that has an associated **object_role.name** =’list of files’, the set of **document_reference_items** pointed to by the **applied_document_reference** shall be constrained to be of type **document_file**. The **document_file** shall have an associated **document_representation_-type.name** =’digital’ or ‘physical’.

5.2.4.33 text_literal_alignment_baseline_constraint

The **text_literal_alignment_baseline_constraint** rule specifies the values for the **text_literal.alignment**.

NOTE This rule applies to mapping table rule 34.

EXPRESS specification:

```
*)
RULE text_literal_alignment_baseline_constraint
   FOR (text_literal);
WHERE
 wr1 : SIZEOF(QUERY(t <* text_literal|
            NOT (t.alignment IN
                 ['left', 'center', 'right']))) = 0;
END_RULE;
(*
```

Argument definitions:

text_literal: identifies the set of all instances of **text_literal** entities.

Formal propositions:

WR1: The values for **text_literal.alignment** shall be 'left', 'center', or 'right'.

5.2.4.34 text_literal_font_requires_externally_defined_text_font

The **text_literal_font_requires_externaly_defined_text_font** rule specifies the value for the **text_-literal.font**.

NOTE This rule applies to mapping table rule 35.

EXPRESS specification:

```
*)
RULE text_literal_font_requires_externally_defined_text_font
   FOR (text_literal);
WHERE
 wr1 : SIZEOF(QUERY(t <* text_literal |
          NOT (('TECHNICAL_DATA_PACKAGING.EXTERNALLY_DEFINED_TEXT_FONT') IN
             TYPEOF (t.font)))) = 0;
END_RULE;
(*
```

Argument definitions:

text_literal: identifies the set of all instances of **text_literal** entities.

Formal propositions:

WR1: The value for **text_literal.font** shall be **externally_defined_text_font**.

5.2.4.35 versioned_action_request_requires_status

The **version_action_request_requires_statu**s rule specifies that each instance of **versioned_action_-request** shall have exactly one **status**. The status of a **versioned_action_reques**t is defined by the **action_-request_status** entity.

Express Specification:

```
*)
RULE versioned_action_request_requires_status FOR
(action_request_status,
versioned_action_request);
WHERE
WR1: SIZEOF ( QUERY ( ar <* versioned_action_request | NOT (
SIZEOF ( QUERY ( ars <* action_request_status | ar:=:ars.
assigned_request ) ) =1 ) ) ) =0;
END_RULE;
(*
```

Argument definitions:

action_request_status: the set of all instances of action_request_status.
versioned_action_request: the set of all instances of versioned_action_request.

Formal propositions:

WR1: For each instance of **versioned_action_request** there shall be exactly one instance of **action_-request_status** that contains an **assigned_request** attribute value to this instance of **versioned_action_-request.**

5.2.5 Technical_data_packaging funtion definitions

5.2.5.1 product_category_graph_members

The **product_category_graph_members** funtion specifies the set of **product_category** instances that are in the same **product_category_relationship**.

EXPRESS specification:

```
*)
FUNCTION product_category_graph_members (p : product_category) :
     SET OF product_category;
LOCAL
  result : SET OF product_category := [];
  pcr : BAG OF product_category_relationship;
END_LOCAL;
  pcr := USEDIN(p, 'TECHNICAL_DATA_PACKAGING.'+
                 'PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY');
  IF (SIZEOF(pcr) > 0) THEN
    result := result + p;
    REPEAT i := 1 TO SIZEOF(pcr);
      result := result +
               product_category_graph_members(pcr[i].sub_category);
    END_REPEAT;
  END_IF;
  pcr := USEDIN(p, 'TECHNICAL_DATA_PACKAGING.'+
                  'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY');
  IF (SIZEOF(pcr) > 0) THEN
```



```
    result := result + p;
    REPEAT i := 1 TO SIZEOF(pcr);
      result := result + product_category_graph_members(pcr[i].category);
    END_REPEAT;
  END_IF;
  RETURN(result);
END_FUNCTION;
(*
```

Argument definitions:

product_category: identifies an instance of a **product_category** entity.

Formal propositions:

Given an instance of a product_category, the function returns the (possibly empty) set of product_category instances that are in the same product_category_relationship graph.

```
*)
END_SCHEMA;
(*
```

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part of ISO 10303, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods:

— ISO 10303-21

Requirements with respect to implementation methods specific requirements are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in annex D.

This part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes:

— Class 1: Data definition exchange (DDE) for files;

— Class 2: Data definition exchange (DDE) for TDP elements;

— Class 3: Data definition exchange (DDE) for indentured methods;

— Class 4: Parts list (PL);

— Class 5: Data list (DL);

— Class 6: Indentured data list (IDL);

— Class 7: Index list (IL);

— Class 8: Other list (OL);

— Class 9: List with presentation;

— Class 10: Reference document identification and drawing identification;

— Class 11: Reference document identification and drawing identification with ISO 10303-201 and ISO 10303-202 drawing presentation identification;

— Class 12: Product data set (PDS) without presentation format;

— Class 13: Product data set (PDS) with shading;

— Class 14: Product data set (PDS) with presentation format;

Support for a particular conformance class requires support of all the options specified in that class.

This part of ISO 10303 specifies fourteen conformance classes. Table 15 identifies the fourteen conformance classes that are defined for this part of ISO 10303.

Table 15 — Conformance classes

Capabilities	Classes													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Data Definition Exchange for files	X								#					
Data Definition Exchange for TDP Elements		X							#					
Data Definition Exchange for Indentured Methods			X						#					
Parts List (PL)				X					#					
Data List (DL)					X				#					
Indentured Data List (IDL)						X			#					
Index List (IDL)							X		#					
Other List (OL)								X	#					
List Presentation									X					

Table 15 — Conformance classes (concluded)

Capabilities	Classes													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Reference Document and Drawing Identification										X	X			
Reference Document and Drawing Presentation Identification with ISO 10303-201 and ISO 10303-202											X			
Product Data Set with surfaces and solids												X	X	X
Shading for surfaces and solids													X	X
Presentation, annotation, and tolerances for Product Data Set														X

Key: X - Must Exist
- List presentation may be directly associated to the data context identified in Conformance Classes 1-8

Conformance to a particular class requires that all AIM elements defined as part of that class be supported. Table 16 defines the classes to which each AIM element belongs.

NOTE 1 ISO TS 10303-332—[3] [5] defines the abstract test suite to be used in the assessment of conformance. ISO 10303-32—[4] [6] describes the conformance assessment process.

NOTE 2 There may be AIM elements that do not belong to any conformance class but appear within the expanded listing by the use of the implicit reference rules of ISO 10303-11. These AIM elements are not required for any implementation of this part of ISO 10303, but were included in Table 16 for completeness.

NOTE 3 For each TDP element and the DDE there are different levels of capability that can be obtained in a data exchange. An implementation that is conformant to one or more of the following conformance classes shall support the highest level of capability. The different levels of capability are described in annex K.

NOTE 4 Files may conform to ISO 10303-21 coding for an ISO 10303-200 series part, raster files, or other formats agreed to by exchanging parties.

EXAMPLE 1 A business practice defines that each TDP element be contained in its own unique file.

[3] To be published.

[4] To be published.

EXAMPLE 2 A business practice defines that a particular collection of TDP elements be contained in a common file.

6.1 Data definition exchange for files, conformance class 1

The Data definition exchange for files has the ability to capture the packaging/shipping information required to identify and relate among one another the files exchanged, in whole or in part, between two enterprises or between two systems. Annex K.5 provides details as to the functionality of this conformance class and provides the list of application objects it supports.

6.2 Data definition exchange for TDP elements, conformance class 2

The Data definition exchange for TDP elements has the ability to capture the packaging/shipping information required to identify and relate among one another the documents and files exchanged, in whole or in part, between two enterprises or between two systems. Annex K.5 provides details as to the functionality of this conformance class and provides the list of application objects it supports.

6.3 Data definition exchange for indentured methods, conformance class 3

The Data definition exchange for indentured methods has the ability to capture the packaging/shipping information required to identify and relate among one another the product structure, documents and files exchanged, in whole or in part, between two enterprises or between two systems. Annex K.5 provides details as to the functionality of this conformance class and provides the list of application objects it supports.

6.4 Parts list, conformance class 4

The Parts list conformance class has the ability to capture part or item information including configuration and accountability information for the purposes of exchanging the bill of material and product structure information between two enterprises or between two systems. These levels of capability do not address graphical presentation of the Parts List. A set of usage options for this conformance class have been established that provide guidance for implementations. The options include a minium level for simple commercial usage, a controlled level where more stringent requirements are needed, a limited history and material identification level, and a supplemental identification level. Annex K.2 provides more details as to the functionality of these conformance class's options and provides the list of application objects it supports.

6.5 Data list, conformance class 5

The Data list has the ability to capture document management and accountability information for the purposes of exchanging a list of document information between two enterprises or between two systems. A set of usage options for this conformance class have been established that provide guidance for implementations. The options include a minium level for simple commercial usage, a controlled level where more stringent requirements are needed, and a supplemental identification level. These levels of capability do not address graphical presentation of the Data List. Annex K.3 provides more details as to the functionality of this conformance class and provides the list of application objects it supports.

6.6 Indentured data list, conformance class 6

The Indentured data list has the ability to capture document management and accountability information for the purposes of exchanging a design document disclosing a master list of Technical Data Package elements (TDP elements) such as engineering drawings, associated lists, referenced documents, or other design disclosure data which together provide product design data for a given product, assembly, or detail. A set of usage options for this conformance class have been established that provide guidance for implementations. The options include a minium level for simple commercial usage, a controlled level where more stringent requirements are needed, a part number indenture basis, and a supplemental identification level. Annex K.6 provides more details as to the functionality of this conformance class and provides the list of application objects it supports.

6.7 Index list, conformance class 7

The Index list has the ability to capture document management and accountability information for the purposes of exchanging a list of data list, indentured data list and other index list information between two enterprises or between two systems. A set of usage options for this conformance class have been established that provide guidance for implementations. The options include a minium level for simple commercial usage, a controlled level where more stringent requirements are needed, and a supplemental identification level. These levels of capability do not address graphical presentation of the Index List. Annex K.4 provides more details as to the functionality of this conformance class and provides the list of application objects it supports.

6.8 Other list, conformance class 8

Other list has the ability to capture the basic table information. Other list will allow the ability to provide row and column information about any product information that needs to be exchanged. Other list provides the same document management and accountability information as the other conformance classes.

6.9 List with presentation, conformance class 9

List with presentation has the ability to capture the presentation information of any document list. The presentation information provides the layout information of each page and the content information. Conformance class nine shall be implemented in conjunction with one or more of conformance classes, one through eight.

6.10 Reference document identification and drawing identification, Conformance Class 10

Reference document identification and drawing identification has the ability to capture the identification of any reference document or drawing with some basic document management and accountability information. The ability to document usage constraints for reference documents and sheet count for drawings is also provided. When this part of ISO 10303 is implemented independently of ISO 10303-201 or ISO 10303-202, the implementation method is defined in Annex C.2.1.

6.11 Reference document identification and drawing identification with ISO 10303-201 and ISO 10303-202 drawing presentation identification, Conformance Class 11

Reference document identification and drawing identification with ISO 10303-201 and ISO 10303-202 drawing presentation identification has the ability to associate the drawing's document management and accountability information to the drawing's content information captured with ISO 10303-201 and ISO 10303-202. The implementation method options are defined in Annex C.2.2 and C.2.3 for this part of ISO 10303 using ISO 10303-505 constructs.

6.12 Product data set (PDS) without presentation format, Conformance Class 12

Product data set (PDS) without presentation format has the ability to capture 3D models with surfaces and solids. This conformance class has the ability to capture document management and accountability information for the purposes of exchanging geometry data sets between two enterprises or between two systems. This conformance class allows geometry models to be managed from a document perspective.

NOTE The lofting surface of a product is usually a collection of surfaces that define the outside target shape of a product. This collection of surfaces are used as the initial / founding shape of the overall airplane. Each mathematical surface in the group can span multiple parts or span only a portion of a part's shape. These groups of surfaces are managed and configured as single units, independent of any single part or assembly. Relationships between parts and these groups of surfaces can be instantiated identifying primary surface requirements for the parts.

EXAMPLE One or more lofting surfaces of an airplane.

The geometry AICs that are options for this conformance class include the following:

— ISO 10303-501, AIC, Edge-based wireframe;

— ISO 10303-502, AIC, Shell-based wireframe;

— ISO 10303-503, AIC, Geometrically bounded 2D wireframe;

— ISO 10303-507, AIC, Geometrically bounded surface;

— ISO 10303-508, AIC, Non-manifold surface;

— ISO 10303-509, AIC, Manifold surface;

— ISO 10303-510, AIC, Geometrically bounded wireframe;

— ISO 10303-511, AIC, Topologically bounded surface;

— ISO 10303-512, AIC, Faceted boundary rrepresentation;

— ISO 10303-513, AIC, Elementary boundary representation;

— ISO 10303-514, AIC, Advanced boundary representation;

— ISO 10303-515, AIC, Constructive solid geometry.

These AIC geometry options can be used together or independently to satisfy this conformance class.

6.13 Product data set (PDS) with shading, conformance class 13

Product data set (PDS) with shading has the ability to capture 3D models with surfaces, solids, and shading. Conformance class 13 is conformance class 12 and shading information. his conformance class has the ability to capture document management and accountability information for the purposes of exchanging geometry data sets between two enterprises or between two systems. AIC, ISO 10303-517, Mechanical design geometric presentation, provides the additinal functionality of conformance class 13 over conformance class 12.

6.14 Product data set (PDS) with presentation format, conformance class 14

Product data set (PDS) with presentation format has the ability to capture 3D models with surfaces, solids, shading, annotation, and presentation information for human readability. Conformance class 14 is conformance class 13 and annotation. This conformance class has the ability to capture document management and accountability information for the purposes of exchanging geometry data sets between two enterprises or between two systems. AIC, ISO 10303-518, Mechanical design shaded presentation, provides the additinal functionality of conformance class 14 over conformance class 13.

Table 16 — Conformance class elements

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
action	X	X	X	X	X	X	X	X		X	X	X	X	X
action_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
action_directive	X	X	X			X								
action_method	X	X	X	X	X	X	X	X		X	X	X	X	X
action_method_with_document_reference	X	X	X	X	X	X	X	X		X	X	X	X	X
action_property				X										
action_property_representation				X										
action_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
action_request_assignment	X	X	X			X								

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
action_request_solution	X	X	X			X								
action_request_status	X	X	X			X								
action_resource	X	X	X			X								
action_resource_relationship	X	X	X			X								
action_resource_type	X	X	X			X								
action_status	X	X	X	X	X	X	X	X		X	X	X	X	X
address	X	X	X	X	X	X	X	X		X	X	X	X	X
advanced_brep_shape_-representation												X	X	X
advanced_face												X	X	X
alternate_product_relationship	X	X	X	X		X						X	X	X
amount_of_substance_-measure_with_unit	X	X	X	X	X	X	X	X		X	X			
amount_of_substance_unit	X	X	X	X	X	X	X	X		X	X			
annotation_curve_occurrence									X					
annotation_occurrence									X			X	X	X
annotation_point_occurrence									X					
annotation_text									X					
annotation_text_character									X					
annotation_text_occurrence									X					
application_context	X	X	X	X	X	X	X	X		X	X	X	X	X
application_context_element	X	X	X	X	X	X	X	X		X	X	X	X	X
application_context_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
application_protocol_definition	X	X	X	X	X	X	X	X	X	X	X	X	X	X
applied_action_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_action_request_-assignment	X	X	X			X								

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
applied_approval_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_certification_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_classification_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_contract_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_date_and_time_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_date_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_document_reference	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_document_usage_-constraint_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_effectivity_assignment	X	X	X	X										
applied_effectivity_context_-assignment	X	X	X	X										
applied_event_occurrence_-assignment	X	X	X	X		X								
applied_external_identification_-assignment	X	X	X									X	X	X
applied_identification_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_organization_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_organizational_project_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_person_and_-organization_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
applied_presented_item									X		X	X	X	X
applied_security_classification_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
approval	X	X	X	X	X	X	X	X		X	X	X	X	X
approval_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
approval_date_time	X	X	X	X	X	X	X	X		X	X	X	X	X
approval_person_organization	X	X	X	X	X	X	X	X		X	X	X	X	X
approval_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
approval_role	X	X	X	X	X	X	X	X		X	X	X	X	X
approval_status	X	X	X	X	X	X	X	X		X	X	X	X	X
area_dependent_annotation_-representation									X					
area_in_set									X					
area_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
area_unit	X	X	X	X	X	X	X	X		X	X			
assembly_component_usage			X	X		X								
assembly_component_usage_-substitute			X	X		X								
assembly_component_usage_-substitute_with_ranking			X	X		X								
axis1_placement												X	X	X
axis2_placement_2d									X			X	X	X
axis2_placement_3d												X	X	X
b_spline_curve												X	X	X
b_spline_curve_with_knots												X	X	X
b_spline_surface												X	X	X
b_spline_surface_with_knots												X	X	X
background_colour													X	X
bezier_curve												X	X	X
bezier_surface												X	X	X
block												X	X	X
boolean_result												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
boundary_curve												X	X	X
bounded_curve												X	X	X
bounded_pcurve												X	X	X
bounded_surface												X	X	X
bounded_surface_curve												X	X	X
box_domain												X	X	X
boxed_half_space												X	X	X
brep_with_voids												X	X	X
calendar_date	X	X	X	X	X	X	X	X		X	X	X	X	X
camera_image									X				X	X
camera_image_3d_with_scale									X					X
camera_model												X	X	X
camera_model_d3													X	X
camera_model_d3_with_hlhsr													X	X
camera_model_with_light_-sources													X	X
camera_usage													X	X
cartesian_point									X			X	X	X
cartesian_transformation_-operator				X								X	X	X
cartesian_transformation_-operator_2d				X								X	X	X
cartesian_transformation_-operator_3d				X								X	X	X
certification	X	X	X	X	X	X	X	X		X	X	X	X	X
certification_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
certification_type	X	X	X	X	X	X	X	X		X	X	X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
character_glyph_style_outline									X					X
character_glyph_style_stroke									X					X
character_glyph_symbol									X					
characterized_object	X	X	X			X						X	X	X
circle												X	X	X
class	X	X	X	X	X	X	X	X		X	X	X	X	X
class_system	X	X	X	X	X	X	X	X		X	X	X	X	X
classification_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
classification_role	X	X	X	X	X	X	X	X		X	X	X	X	X
closed_shell												X	X	X
colour													X	X
colour_rgb													X	X
colour_specification													X	X
composite_curve												X	X	X
composite_curve_on_surface												X	X	X
composite_curve_segment												X	X	X
composite_text									X					
compound_representation_item								X				X	X	X
configuration_design	X	X	X	X	X	X	X	X		X	X	X	X	X
configuration_effectivity	X	X	X	X	X	X	X	X		X	X			
configuration_item	X	X	X	X	X	X	X	X		X	X	X	X	X
conic												X	X	X
conical_surface												X	X	X
connected_edge_set												X	X	X
connected_face_set												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
context_dependent_shape_-representation				X										
context_dependent_unit	X	X	X	X	X	X	X	X		X	X			
contract	X	X	X	X	X	X	X	X		X	X			
contract_assignment	X	X	X	X	X	X	X	X		X	X			
contract_relationship	X	X	X	X	X	X	X	X		X	X			
contract_type	X	X	X	X	X	X	X	X		X	X			
conversion_based_unit	X	X	X	X	X	X	X	X		X	X			
coordinated_universal_time_-offset	X	X	X	X	X	X	X	X		X	X	X	X	X
csg_shape_representation												X	X	X
csg_solid												X	X	X
curve												X	X	X
curve_bounded_surface												X	X	X
curve_replica												X	X	X
curve_style													X	X
curve_style_font														X
curve_style_font_pattern														X
curve_style_rendering													X	X
cylindrical_surface												X	X	X
date	X	X	X	X	X	X	X	X		X	X	X	X	X
date_and_time	X	X	X	X	X	X	X	X		X	X	X	X	X
date_and_time_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
date_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
date_role	X	X	X	X	X	X	X	X		X	X	X	X	X
date_time_role	X	X	X	X	X	X	X	X		X	X	X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
dated_effectivity			X	X		X								
defined_character_glyph									X					
definitional_representation												X	X	X
degenerate_pcurve												X	X	X
degenerate_toroidal_surface												X	X	X
derived_unit	X	X	X	X	X	X	X	X		X	X			
derived_unit_element	X	X	X	X	X	X	X	X		X	X			
description_attribute	X	X	X	X	X	X	X	X		X	X	X	X	X
descriptive_representation_item	X	X	X	X	X	X	X	X		X	X	X	X	X
design_make_from_relationship			X	X		X								
dimensional_exponents				X										
directed_action	X	X	X			X								
direction												X	X	X
document	X	X	X	X	X	X	X	X		X	X	X	X	X
document_file	X	X	X	X								X	X	X
document_product_association	X	X	X	X	X	X	X	X		X	X	X	X	X
document_product_equivalence	X	X	X	X	X	X	X	X		X	X	X	X	X
document_reference	X	X	X	X	X	X	X	X		X	X	X	X	X
document_relationship				X		X	X			X	X	X	X	X
document_representation_type	X	X	X	X	X	X	x	X		X	X	X	X	X
document_type	X	X	X	X	X	X	X	X		X	X	X	X	X
document_usage_constraint	X	X	X	X	X	X	X	X		X	X	X	X	X
document_usage_constraint_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
document_usage_role	X	X	X	X	X	X	X	X		X	X	X	X	X
draughting_approval_assignment											X			

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
draughting_callout											X			X
draughting_drawing_revision				X	X			X			X			
draughting_pre_defined_colour														X
draughting_pre_defined_curve_-font														X
draughting_title											X			X
drawing_definition											X			
drawing_revision											X			
drawing_sheet_revision											X			
edge												X	X	X
edge_based_wireframe_model												X	X	X
edge_based_wireframe_shape_-representation												X	X	X
edge_curve												X	X	X
edge_loop												X	X	X
effectivity			X	X		X								
effectivity_assignment			X	X		X								
effectivity_context_assignment			X	X		X								
effectivity_context_role			X	X		X								
effectivity_relationship			X	X		X								
electric_current_measure_with_-unit	X	X	X	X	X	X	X	X		X	X			
electric_current_unit	X	X	X	X	X	X	X	X		X	X			
elementary_brep_shape_-representation												X	X	X
elementary_surface												X	X	X
ellipse												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
evaluated_degenerate_pcurve												X	X	X
event_occurrence	X	X	X	X		X								
event_occurrence_assignment	X	X	X	X		X								
event_occurrence_relationship	X	X	X	X		X								
event_occurrence_role	X	X	X	X		X								
executed_action	X	X	X	X	X	X	X	X		X	X	X	X	X
external_identification_-assignment	X	X	X						X					
external_source	X	X	X						X					
externally_defined_character_-glyph									X					
externally_defined_class	X	X	X	X		X								
externally_defined_general_-property	X	X	X	X		X								
externally_defined_item									X					
externally_defined_planar_box									X					
externally_defined_style									X					
externally_defined_symbol									X					
externally_defined_symbol_-and_placement									X					
externally_defined_text_font									X					
extruded_face_solid												X	X	X
face												X	X	X
face_based_surface_model												X	X	X
face_bound												X	X	X
face_outer_bound												X	X	X
face_surface												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
faceted_brep												X	X	X
faceted_brep_shape_-representation												X	X	X
fill_area_style														X
fill_area_style_colour														X
founded_item												X	X	X
functionally_defined_-transformation												X	X	X
general_property	X	X	X	X		X						X	X	X
general_property_association	X	X	X	X		X						X	X	X
general_property_relationship	X	X	X	X		X						X	X	X
geometric_curve_set												X	X	X
geometric_representation_-context									X			X	X	X
geometric_representation_item									X			X	X	X
geometric_set												X	X	X
geometrically_bounded_2d_-wireframe_representation												X	X	X
geometrically_bounded_-surface_shape_representation												X	X	X
geometrically_bounded_-wireframe_shape_representation												X	X	X
global_uncertainty_assigned_-context	X	X	X	X							X	X	X	X
global_unit_assigned_context									X			X	X	X
group	X	X	X	X	X	X	X	X		X	X			
group_relationship	X	X	X	X	X	X	X	X		X	X			
half_space_solid												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
hyperbola												X	X	X
id_attribute	X	X	X	X	X	X	X	X		X	X	X	X	X
identification_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
identifcation_assignment_-relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
identification_role	X	X	X	X	X	X	X	X		X	X	X	X	X
intersection_curve												X	X	X
invisibility														X
item_defined_transformation												X	X	X
item_identified_representation_-usage												X	X	X
language_assignment	X	X	X	X	X	X	X	X		X	X			X
length_measure_with_unit	X	X	X	X	X	X	X	X	X	X	X			
length_unit	X	X	X	X	X	X	X	X	X	X	X			
library_context									X					
light_source													X	X
light_source_ambient													X	X
light_source_direction													X	X
light_source_positional													X	X
light_source_spot													X	X
line									X			X	X	X
local_time	X	X	X	X	X	X	X	X		X	X	X	X	X
loop												X	X	X
lot_effectivity			X	X		X								
luminous_intensity_measure_-with_unit	X	X	X	X	X	X	X	X		X	X			
luminous_intensity_unit	X	X	X	X	X	X	X	X		X	X			

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
make_from_usage_option			X	X		X								
make_from_usage_option_-with_reference_designator			X	X		X								
manifold_solid_brep												X	X	X
manifold_surface_shape_-representation												X	X	X
mapped_item								X	X			X	X	X
mass_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
mass_unit	X	X	X	X	X	X	X	X		X	X			
material_designation				X										
measure_qualification				X										
measure_representation_item	X	X	X	X	X	X	X	X		X	X			
measure_with_unit	X	X	X	X	X	X	X	X		X	X			
mechanical_design_geometric_-presentation_area													X	X
mechanical_design_geometric_-presentation_representation													X	X
mechanical_design_shaded_-presentation_area													X	X
mechanical_design_shaded_-presentation_representation													X	X
name_attribute	X	X	X	X	X	X	X	X		X	X	X	X	X
named_unit	X	X	X	X	X	X	X	X	X	X	X			
next_assembly_usage_-occurrence			X	X		X								
non-manifold_surface_shape_-representation												X	X	X
object_role	X	X	X	X	X	X	X	X		X	X	X	X	X
offset_curve_2d												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
offset_curve_3d												X	X	X
offset_surface												X	X	X
open_shell												X	X	X
organization	X	X	X	X	X	X	X	X		X	X	X	X	X
organization_assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
organization_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
organization_role	X	X	X	X	X	X	X	X		X	X	X	X	X
organizational_address	X	X	X	X	X	X	X	X		X	X	X	X	X
organizational_project	X	X	X	X	X	X	X	X		X	X	X	X	X
organizational_project_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
organizational_project_-relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
organizational_project_role	X	X	X	X	X	X	X	X		X	X	X	X	X
oriented_closed_shell												X	X	X
oriented_edge												X	X	X
oriented_face												X	X	X
oriented_open_shell												X	X	X
oriented_path												X	X	X
oriented_surface												X	X	X
other_list_table_representation								X						
outer_boundary_curve												X	X	X
over_riding_style_item									X					X
parabola												X	X	X
parametric_representation_-context												X	X	X
path												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
pcurve												X	X	X
person	X	X	X	X	X	X	X	X		X	X	X	X	X
person_and_organization	X	X	X	X	X	X	X	X		X	X	X	X	X
person_and_organization_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
person_and_organization_role	X	X	X	X	X	X	X	X		X	X	X	X	X
personal_address	X	X	X	X	X	X	X	X		X	X			
placement									X			X	X	X
planar_box									X			X	X	X
planar_extent									X			X	X	X
plane									X			X	X	X
plane_angle_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
plane_angle_unit	X	X	X	X	X	X	X	X		X	X			
point									X			X	X	X
point_on_curve												X	X	X
point_on_surface												X	X	X
point_replica												X	X	X
point_style													X	X
poly_loop												X	X	X
polyline												X	X	X
pre_defined_character_glyph									X					
pre_defined_colour													X	X
pre_defined_curve_font														X
pre_defined_item													X	X
pre_defined_text_font									X					X
presentation_area									X					X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
precision_qualifier				X										
presentation_representation									X				X	X
presentation_set									X		X			X
presentation_size									X				X	X
presentation_style_assignment													X	X
presentation_style_by_context													X	X
presentation_view													X	X
presented_item									X		X	X	X	X
presented_item_representation									X		X	X	X	X
process_product_association	X	X	X	X	X	X	X	X		X	X	X	X	X
product	X	X	X	X	X	X	X	X		X	X	X	X	X
product_category	X	X	X	X	X	X	X	X		X	X	X	X	X
product_category_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
product_concept	X	X	X	X	X	X	X	X		X	X	X	X	X
product_concept_context	X	X	X	X	X	X	X	X		X	X	X	X	X
product_context	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_context	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_context_-association	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_context_role	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_effectivity	X	X	X	X	X	X	X	X		X	X			
product_definition_formation	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_formation_-relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_formation_-with_specified_source	X	X	X	X	X	X	X	X		X	X	X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
product_definition_occurrence_-relationship			X	X		X								
product_definition_process	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
product_definition_shape			X			X			X			X	X	X
product_definition_substitute			X	X		X								
product_definition_usage		X	X	X		X								
product_definition_with_-associated_documents	X	X	X	X	X	X	X	X		X	X	X	X	X
product_related_product_-category	X	X	X	X	X	X	X	X		X	X	X	X	X
product_relationship	X	X	X	X	X	X	X	X		X	X	X	X	X
promissory_usage_occurrence		X	X	X		X								
property_definition	X	X	X	X	X	X	X	X		X	X	X	X	X
property_definition_relationship												X		X
property_definition_-representation	X	X	X	X	X	X	X	X		X	X	X	X	X
qualified_representation_item				X										
quantified_assembly_-component_usage			X	X		X								
quasi_uniform_curve												X	X	X
quasi_uniform_surface												X	X	X
ratio_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
ratio_unit	X	X	X	X	X	X	X	X		X	X			
rational_b_spline_curve												X	X	X
rational_b_spline_surface												X	X	X
rectangular_composite_surface												X	X	X
rectangular_trimmed_surface												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
relative_event_occurrence	X	X	X	X		X								
reparametrised_composite_-curve_segment												X	X	X
representation	X	X	X	X	X	X	X	X	X	X	X	X	X	X
representation_context	X	X	X	X	X	X	X	X	X	X	X	X	X	X
representation_item	X	X	X	X	X	X	X	X	X	X	X	X	X	X
representation_item_relationship								X				X	X	X
representation_map								X	X			X	X	X
representation_relationship								X						
representation_relationship_-with_transformation				X										
revolved_face_solid												X	X	X
right_angular_wedge												X	X	X
right_circular_cone												X	X	X
right_circular_cylinder												X	X	X
role_association	X	X	X	X	X	X	X	X		X	X	X	X	X
seam_curve												X	X	X
security_classification	X	X	X	X	X	X	X	X		X	X	X	X	X
security_classification_-assignment	X	X	X	X	X	X	X	X		X	X	X	X	X
security_classification_level	X	X	X	X	X	X	X	X		X	X	X	X	X
serial_numbered_effectivity			X	X		X								
shape_aspect												X	X	X
shape_aspect_relationship												X	X	X
shape_definition_representation	X	X	X	X										
shape_representation			X			X						X	X	X
shell_based_surface_model												X	X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
shell_based_wireframe_model												X	X	X
shell_based_wireframe_shape_-representation												X	X	X
si_unit	X	X	X	X	X	X	X	X	X	X	X	X	X	X
solid_angle_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
solid_angle_unit	X	X	X	X	X	X	X	X		X	X			
solid_model												X	X	X
solid_replica												X	X	X
specified_high_usage_-occurrence	X	X	X	X		X						X	X	X
sphere												X	X	X
spherical_surface												X	X	X
styled_item									X				X	X
surface												X	X	X
surface_curve												X	X	X
surface_of_linear_extrusion												X	X	X
surface_of_revolution												X	X	X
surface_patch												X	X	X
surface_rendering_properties													X	X
surface_replica												X	X	X
surface_side_style													X	X
surface_style_boundary													X	X
surface_style_control_grid														X
surface_style_fill_area														X
surface_style_parameter_line														X
surface_style_reflectance_-ambient													X	X

Table 16 — Conformance class elements (continued)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
surface_style_reflectance_-ambient_diffuse													X	X
surface_style_reflectance_-ambient_diffuse_specular													X	X
surface_style_rendering													X	X
surface_style_rendering_with_-properties													X	X
surface_style_segmentation_-curve													X	X
surface_style_silhouette														X
surface_style_transparent												X	X	X
surface_style_usage												X	X	X
swept_face_solid												X	X	X
swept_surface												X	X	X
symbol_representation									X					
text_literal									X					
text_string_representation									X					
text_style									X					X
text_style_for_defined_font									X					X
thermodynamic_temperature_-measure_with_unit	X	X	X	X	X	X	X	X		X	X			
thermodynamic_temperature_-unit	X	X	X	X	X	X	X	X		X	X			
time_interval	X	X	X	X		X								
time_interval_based_effectivity	X	X	X	X		X								
time_interval_relationship	X	X	X	X		X								
time_interval_with_bounds	X	X	X	X		X								
time_measure_with_units	X	X	X	X	X	X	X	X		X	X			

Table 16 — Conformance class elements (concluded)

AIM element	Class													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
time_unit	X	X	X	X	X	X	X	X		X	X			
topological_representation_item												X	X	X
toroidal_surface												X	X	X
torus												X	X	X
trimmed_curve												X	X	X
type_qualifier				X										
uncertainty_measure_with_unit	X	X	X	X							X	X	X	X
uniform_curve												X	X	X
uniform_surface												X	X	X
value_representation_item	X	X	X	X	X	X	X	X						
vector												X	X	X
versioned_action_request	X	X	X			X								
vertex												X	X	X
vertex_loop												X	X	X
vertex_point												X	X	X
vertex_shell												X	X	X
view_volume				X										
volume_measure_with_unit	X	X	X	X	X	X	X	X		X	X			
volume_unit	X	X	X	X	X	X	X	X		X	X			
wire_shell												X	X	X

Annex A
(normative)

AIM EXPRESS expanded listing

The following EXPRESS is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used.

```
SCHEMA technical_data_packaging;

  CONSTANT
  dummy_gri : geometric_representation_item := representation_item('') ||
                  geometric_representation_item();
  dummy_tri : topological_representation_item := representation_item('')
                  || topological_representation_item();
  END_CONSTANT;

  TYPE action_item = SELECT
    (configuration_effectivity,
     document_file,
     product_definition,
     product_definition_with_associated_documents,
     product_definition_relationship,
     product_definition_formation,
     representation);
  END_TYPE; -- action_item

  TYPE action_request_item = SELECT
    (product_definition,
     product_definition_formation,
     product_definition_relationship,
     property_definition);
  END_TYPE; -- action_request_item

  TYPE ahead_or_behind = ENUMERATION OF
    (ahead,
     exact,
     behind);
  END_TYPE; -- ahead_or_behind

  TYPE amount_of_substance_measure = REAL;
  END_TYPE; -- amount_of_substance_measure

  TYPE approval_item = SELECT
    (action,
     applied_action_assignment,
     applied_certification_assignment,
     applied_effectivity_assignment,
     certification,
     configuration_effectivity,
     configuration_item,
     contract,
     document_file,
     product_definition,
     product_definition_formation,
     product_definition_relationship,
```

```
    property_definition,
    versioned_action_request);
END_TYPE; -- approval_item

TYPE approved_item = SELECT
  (drawing_revision,
   drawing_sheet_revision);
END_TYPE; -- approved_item

TYPE area_measure = REAL;
END_TYPE; -- area_measure

TYPE area_or_view = SELECT
  (presentation_area,
   presentation_view);
END_TYPE; -- area_or_view

TYPE axis2_placement = SELECT
  (axis2_placement_2d,
   axis2_placement_3d);
END_TYPE; -- axis2_placement

TYPE b_spline_curve_form = ENUMERATION OF
  (polyline_form,
   circular_arc,
   elliptic_arc,
   parabolic_arc,
   hyperbolic_arc,
   unspecified);
END_TYPE; -- b_spline_curve_form

TYPE b_spline_surface_form = ENUMERATION OF
  (plane_surf,
   cylindrical_surf,
   conical_surf,
   spherical_surf,
   toroidal_surf,
   surf_of_revolution,
   ruled_surf,
   generalised_cone,
   quadric_surf,
   surf_of_linear_extrusion,
   unspecified);
END_TYPE; -- b_spline_surface_form

TYPE boolean_operand = SELECT
  (solid_model,
   half_space_solid,
   csg_primitive,
   boolean_result);
END_TYPE; -- boolean_operand

TYPE boolean_operator = ENUMERATION OF
  (union,
   intersection,
   difference);
END_TYPE; -- boolean_operator
```



```
TYPE celsius_temperature_measure = REAL;
END_TYPE; -- celsius_temperature_measure

TYPE central_or_parallel = ENUMERATION OF
  (central,
   parallel);
END_TYPE; -- central_or_parallel

TYPE certification_item = SELECT
  (product_definition_formation_relationship);
END_TYPE; -- certification_item

TYPE character_style_select = SELECT
  (character_glyph_style_stroke,
   character_glyph_style_outline,
   text_style_for_defined_font);
END_TYPE; -- character_style_select

TYPE characterized_action_definition = SELECT
  (action,
   action_method,
   action_relationship);
END_TYPE; -- characterized_action_definition

TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition

TYPE characterized_product_definition = SELECT
  (product_definition,
   product_definition_relationship);
END_TYPE; -- characterized_product_definition

TYPE classification_item = SELECT
  (action_status,
   approval,
   descriptive_representation_item,
   product_definition);
END_TYPE; -- classification_item

TYPE compound_item_definition = SELECT
  (list_representation_item);
END_TYPE; -- compound_item_definition

TYPE configuration_design_item = SELECT
  (product_definition,
   product_definition_formation);
END_TYPE; -- configuration_design_item

TYPE context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure

TYPE contract_item = SELECT
  (action,
   product_definition,
   product_definition_formation,
   product_definition_with_associated_documents,
```

```
    property_definition);
END_TYPE; -- contract_item

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure

TYPE csg_primitive = SELECT
  (sphere,
   block,
   right_angular_wedge,
   torus,
   right_circular_cone,
   right_circular_cylinder);
END_TYPE; -- csg_primitive

TYPE csg_select = SELECT
  (boolean_result,
   csg_primitive);
END_TYPE; -- csg_select

TYPE curve_font_or_scaled_curve_font_select = SELECT
  (curve_style_font_select);
END_TYPE; -- curve_font_or_scaled_curve_font_select

TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve,
   composite_curve_on_surface);
END_TYPE; -- curve_on_surface

TYPE curve_or_render = SELECT
  (curve_style,
   curve_style_rendering);
END_TYPE; -- curve_or_render

TYPE curve_style_font_select = SELECT
  (curve_style_font,
   pre_defined_curve_font);
END_TYPE; -- curve_style_font_select

TYPE date_and_time_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   event_occurrence,
   organizational_project,
   product_definition,
   product_definition_formation,
   representation,
   security_classification,
   versioned_action_request);
END_TYPE; -- date_and_time_item
```



```
TYPE date_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   event_occurrence,
   organizational_project,
   product_definition,
   product_definition_formation,
   representation,
   security_classification,
   versioned_action_request);
END_TYPE; -- date_item

TYPE date_time_or_event_occurrence = SELECT
  (date_time_select,
   event_occurrence);
END_TYPE; -- date_time_or_event_occurrence

TYPE date_time_select = SELECT
  (date,
   local_time,
   date_and_time);
END_TYPE; -- date_time_select

TYPE day_in_month_number = INTEGER;
WHERE
  wr1: ((1 <= SELF) AND (SELF <= 31));
END_TYPE; -- day_in_month_number

TYPE defined_glyph_select = SELECT
  (pre_defined_character_glyph,
   externally_defined_character_glyph);
END_TYPE; -- defined_glyph_select

TYPE defined_symbol_select = SELECT
  (externally_defined_symbol);
END_TYPE; -- defined_symbol_select

TYPE derived_property_select = SELECT
  (property_definition,
   action_property);
END_TYPE; -- derived_property_select

TYPE description_attribute_select = SELECT
  (action_request_solution,
   application_context,
   approval_role,
   configuration_design,
   date_role,
   date_time_role,
   context_dependent_shape_representation,
   effectivity,
   external_source,
   organization_role,
   person_and_organization_role,
```

```
    person_and_organization,
    property_definition_representation,
    representation);
END_TYPE; -- description_attribute_select

TYPE descriptive_measure = STRING;
END_TYPE; -- descriptive_measure

TYPE dimension_count = INTEGER;
WHERE
  wr1: (SELF > 0);
END_TYPE; -- dimension_count

TYPE direction_count_select = SELECT
  (u_direction_count,
   v_direction_count);
END_TYPE; -- direction_count_select

TYPE document_reference_item = SELECT
  (action,
   action_method,
   alternate_product_relationship,
   applied_document_reference,
   assembly_component_usage_substitute,
   descriptive_representation_item,
   document_file,
   document_relationship,
   executed_action,
   externally_defined_symbol_and_placement,
   make_from_usage_option,
   make_from_usage_option_with_reference_designator,
   next_assembly_usage_occurrence,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_occurrence_relationship,
   product_definition_relationship,
   product_definition_with_associated_documents,
   promissory_usage_occurrence,
   property_definition,
   quantified_assembly_component_usage,
   shape_aspect,
   shape_aspect_relationship,
   specified_higher_usage_occurrence,
   versioned_action_request);
END_TYPE; -- document_reference_item

TYPE draughting_callout_element = SELECT
  (annotation_text_occurrence,
   annotation_curve_occurrence);
END_TYPE; -- draughting_callout_element

TYPE draughting_titled_item = SELECT
  (drawing_revision,
   drawing_sheet_revision);
END_TYPE; -- draughting_titled_item
```



```
TYPE effectivity_context_item = SELECT
  (organization,
   product_definition_formation);
END_TYPE; -- effectivity_context_item

TYPE effectivity_item = SELECT
  (product_definition,
   product_definition_formation,
   product_definition_relationship);
END_TYPE; -- effectivity_item

TYPE electric_current_measure = REAL;
END_TYPE; -- electric_current_measure

TYPE event_occurrence_item = SELECT
  (organizational_project);
END_TYPE; -- event_occurrence_item

TYPE external_identification_item = SELECT
  (document_file,
   product_definition);
END_TYPE; -- external_identification_item

TYPE fill_style_select = SELECT
  (fill_area_style_colour);
END_TYPE; -- fill_style_select

TYPE font_select = SELECT
  (pre_defined_text_font,
   externally_defined_text_font);
END_TYPE; -- font_select

TYPE founded_item_select = SELECT
  (founded_item,
   representation_item);
END_TYPE; -- founded_item_select

TYPE geometric_set_select = SELECT
  (point,
   curve,
   surface);
END_TYPE; -- geometric_set_select

TYPE hour_in_day = INTEGER;
WHERE
  wr1: ((0 <= SELF) AND (SELF < 24));
END_TYPE; -- hour_in_day

TYPE id_attribute_select = SELECT
  (action,
   address,
   product_category,
   property_definition,
   shape_aspect,
   shape_aspect_relationship,
   application_context,
   group,
```

```
    organizational_project,
    representation);
  END_TYPE; -- id_attribute_select

  TYPE identification_item = SELECT
    (application_context,
    applied_document_reference,
    applied_document_usage_constraint_assignment,
    approval_status,
    document_file,
    organization,
    product,
    product_definition,
    product_definition_formation,
    security_classification_level,
    shape_aspect_relationship);
  END_TYPE; -- identification_item

  TYPE identifier = STRING;
  END_TYPE; -- identifier

  TYPE invisible_item = SELECT
    (styled_item,
    representation);
  END_TYPE; -- invisible_item

  TYPE knot_type = ENUMERATION OF
    (uniform_knots,
    quasi_uniform_knots,
    piecewise_bezier_knots,
    unspecified);
  END_TYPE; -- knot_type

  TYPE label = STRING;
  END_TYPE; -- label

  TYPE language_item = SELECT
    (representation);
  END_TYPE; -- language_item

  TYPE layered_item = SELECT
    (presentation_representation,
    representation_item);
  END_TYPE; -- layered_item

  TYPE length_measure = REAL;
  END_TYPE; -- length_measure

  TYPE list_of_reversible_topology_item = LIST [0:?] OF
             reversible_topology_item;
  END_TYPE; -- list_of_reversible_topology_item

  TYPE list_representation_item = LIST [1:?] OF representation_item;
  END_TYPE; -- list_representation_item

  TYPE luminous_intensity_measure = REAL;
  END_TYPE; -- luminous_intensity_measure
```



```
TYPE marker_select = SELECT
  (marker_type);
END_TYPE; -- marker_select

TYPE marker_type = ENUMERATION OF
  (dot,
   x,
   plus,
   asterisk,
   ring,
   square,
   triangle);
END_TYPE; -- marker_type

TYPE mass_measure = REAL;
END_TYPE; -- mass_measure

TYPE measure_value = SELECT
  (length_measure,
   mass_measure,
   time_measure,
   electric_current_measure,
   thermodynamic_temperature_measure,
   celsius_temperature_measure,
   amount_of_substance_measure,
   luminous_intensity_measure,
   plane_angle_measure,
   solid_angle_measure,
   area_measure,
   volume_measure,
   ratio_measure,
   parameter_value,
   numeric_measure,
   context_dependent_measure,
   descriptive_measure,
   positive_length_measure,
   positive_plane_angle_measure,
   positive_ratio_measure,
   count_measure);
END_TYPE; -- measure_value

TYPE message = STRING;
END_TYPE; -- message

TYPE minute_in_hour = INTEGER;
WHERE
  wr1: ((0 <= SELF) AND (SELF <= 59));
END_TYPE; -- minute_in_hour

TYPE month_in_year_number = INTEGER;
WHERE
  wr1: ((1 <= SELF) AND (SELF <= 12));
END_TYPE; -- month_in_year_number

TYPE name_attribute_select = SELECT
  (action_request_solution,
   address,
   configuration_design,
   context_dependent_shape_representation,
```

```
    derived_unit,
    effectivity,
    person_and_organization,
    product_definition,
    product_definition_substitute,
    property_definition_representation);
END_TYPE; -- name_attribute_select

TYPE numeric_measure = NUMBER;
END_TYPE; -- numeric_measure

TYPE organization_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_identification_assignment,
   configuration_item,
   contract,
   document_file,
   event_occurrence,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   security_classification,
   versioned_action_request);
END_TYPE; -- organization_item

TYPE organizational_project_item = SELECT
  (action,
   product_concept);
END_TYPE; -- organizational_project_item

TYPE parameter_value = REAL;
END_TYPE; -- parameter_value

TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE; -- pcurve_or_surface

TYPE person_and_organization_item = SELECT
  (action,
   applied_action_assignment,
   applied_contract_assignment,
   applied_identification_assignment,
   configuration_item,
   contract,
   document_file,
   event_occurrence,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   security_classification,
   versioned_action_request);
END_TYPE; -- person_and_organization_item
```



```
TYPE person_organization_select = SELECT
  (person,
   organization,
   person_and_organization);
END_TYPE; -- person_organization_select

TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure

TYPE positive_length_measure = length_measure;
WHERE
  wr1: (SELF > 0);
END_TYPE; -- positive_length_measure

TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
  wr1: (SELF > 0);
END_TYPE; -- positive_plane_angle_measure

TYPE positive_ratio_measure = ratio_measure;
WHERE
  wr1: (SELF > 0);
END_TYPE; -- positive_ratio_measure

TYPE preferred_surface_curve_representation = ENUMERATION OF
  (curve_3d,
   pcurve_s1,
   pcurve_s2);
END_TYPE; -- preferred_surface_curve_representation

TYPE presentable_text = STRING;
END_TYPE; -- presentable_text

TYPE presentation_representation_select = SELECT
  (presentation_representation,
   presentation_set);
END_TYPE; -- presentation_representation_select

TYPE presentation_size_assignment_select = SELECT
  (presentation_view,
   presentation_area,
   area_in_set);
END_TYPE; -- presentation_size_assignment_select

TYPE presentation_style_select = SELECT
  (point_style,
   curve_style,
   surface_style_usage,
   fill_area_style,
   text_style,
   externally_defined_style);
END_TYPE; -- presentation_style_select

TYPE presented_item_item = SELECT
  (product_definition_formation);
END_TYPE; -- presented_item_item
```

```
TYPE product_or_formation_or_definition = SELECT
  (product,
   product_definition_formation,
   product_definition);
END_TYPE; -- product_or_formation_or_definition

TYPE ratio_measure = REAL;
END_TYPE; -- ratio_measure

TYPE rendering_properties_select = SELECT
  (surface_style_reflectance_ambient,
   surface_style_transparent);
END_TYPE; -- rendering_properties_select

TYPE represented_definition = SELECT
  (general_property,
   property_definition,
   property_definition_relationship,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE; -- represented_definition

TYPE reversible_topology = SELECT
  (reversible_topology_item,
   list_of_reversible_topology_item,
   set_of_reversible_topology_item);
END_TYPE; -- reversible_topology

TYPE reversible_topology_item = SELECT
  (edge,
   path,
   face,
   face_bound,
   closed_shell,
   open_shell);
END_TYPE; -- reversible_topology_item

TYPE role_select = SELECT
  (action_assignment,
   action_request_assignment,
   approval_assignment,
   approval_date_time,
   certification_assignment,
   contract_assignment,
   document_reference,
   effectivity_assignment,
   security_classification_assignment);
END_TYPE; -- role_select

TYPE second_in_minute = REAL;
WHERE
  wr1: ((0 <= SELF) AND (SELF <= 60));
END_TYPE; -- second_in_minute

TYPE security_classification_item = SELECT
  (document_file,
   product_definition,
```



```
   product_definition_formation,
   product_definition_relationship,
   property_definition);
END_TYPE; -- security_classification_item

TYPE set_of_reversible_topology_item = SET [0:?] OF
           reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_item

TYPE shading_curve_method = ENUMERATION OF
  (constant_colour,
   linear_colour);
END_TYPE; -- shading_curve_method

TYPE shading_surface_method = ENUMERATION OF
  (constant_shading,
   colour_shading,
   dot_shading,
   normal_shading);
END_TYPE; -- shading_surface_method

TYPE shape_definition = SELECT
  (product_definition_shape,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE; -- shape_definition

TYPE shell = SELECT
  (vertex_shell,
   wire_shell,
   open_shell,
   closed_shell);
END_TYPE; -- shell

TYPE si_prefix = ENUMERATION OF
  (exa,
   peta,
   tera,
   giga,
   mega,
   kilo,
   hecto,
   deca,
   deci,
   centi,
   milli,
   micro,
   nano,
   pico,
   femto,
   atto);
END_TYPE; -- si_prefix

TYPE si_unit_name = ENUMERATION OF
  (metre,
   gram,
   second,
   ampere,
   kelvin,
```

```
    mole,
    candela,
    radian,
    steradian,
    hertz,
    newton,
    pascal,
    joule,
    watt,
    coulomb,
    volt,
    farad,
    ohm,
    siemens,
    weber,
    tesla,
    henry,
    degree_celsius,
    lumen,
    lux,
    becquerel,
    gray,
    sievert);
END_TYPE; -- si_unit_name

TYPE size_select = SELECT
  (positive_length_measure,
   measure_with_unit,
   descriptive_measure);
END_TYPE; -- size_select

TYPE solid_angle_measure = REAL;
END_TYPE; -- solid_angle_measure

TYPE source = ENUMERATION OF
  (made,
   bought,
   not_known);
END_TYPE; -- source

TYPE source_item = SELECT
  (identifier,
   message);
END_TYPE; -- source_item

TYPE style_context_select = SELECT
  (group,
   representation,
   representation_item,
   presentation_set);
END_TYPE; -- style_context_select

TYPE supported_item = SELECT
  (action_directive,
   action,
   action_method);
END_TYPE; -- supported_item
```



```
TYPE surface_boundary = SELECT
  (boundary_curve,
   degenerate_pcurve);
END_TYPE; -- surface_boundary

TYPE surface_model = SELECT
  (shell_based_surface_model,
   face_based_surface_model);
END_TYPE; -- surface_model

TYPE surface_side = ENUMERATION OF
  (positive,
   negative,
   both);
END_TYPE; -- surface_side

TYPE surface_side_style_select = SELECT
  (surface_side_style);
END_TYPE; -- surface_side_style_select

TYPE surface_style_element_select = SELECT
  (surface_style_fill_area,
   surface_style_boundary,
   surface_style_silhouette,
   surface_style_segmentation_curve,
   surface_style_control_grid,
   surface_style_parameter_line,
   surface_style_rendering);
END_TYPE; -- surface_style_element_select

TYPE text = STRING;
END_TYPE; -- text

TYPE text_alignment = label;
END_TYPE; -- text_alignment

TYPE text_or_character = SELECT
  (annotation_text,
   annotation_text_character,
   defined_character_glyph,
   composite_text,
   text_literal);
END_TYPE; -- text_or_character

TYPE text_path = ENUMERATION OF
  (left,
   right,
   up,
   down);
END_TYPE; -- text_path

TYPE thermodynamic_temperature_measure = REAL;
END_TYPE; -- thermodynamic_temperature_measure

TYPE time_measure = REAL;
END_TYPE; -- time_measure
```

```
TYPE transformation = SELECT
  (item_defined_transformation,
   functionally_defined_transformation);
END_TYPE; -- transformation

TYPE transition_code = ENUMERATION OF
  (discontinuous,
   continuous,
   cont_same_gradient,
   cont_same_gradient_same_curvature);
END_TYPE; -- transition_code

TYPE trimming_preference = ENUMERATION OF
  (cartesian,
   parameter,
   unspecified);
END_TYPE; -- trimming_preference

TYPE trimming_select = SELECT
  (cartesian_point,
   parameter_value);
END_TYPE; -- trimming_select

TYPE u_direction_count = INTEGER;
WHERE
  wr1: (SELF > 1);
END_TYPE; -- u_direction_count

TYPE unit = SELECT
  (named_unit,
   derived_unit);
END_TYPE; -- unit

TYPE v_direction_count = INTEGER;
WHERE
  wr1: (SELF > 1);
END_TYPE; -- v_direction_count

TYPE value_qualifier = SELECT
  (precision_qualifier,
   type_qualifier);
END_TYPE; -- value_qualifier

TYPE vector_or_direction = SELECT
  (vector,
   direction);
END_TYPE; -- vector_or_direction

TYPE volume_measure = REAL;
END_TYPE; -- volume_measure

TYPE wireframe_model = SELECT
  (shell_based_wireframe_model,
   edge_based_wireframe_model);
END_TYPE; -- wireframe_model

TYPE year_number = INTEGER;
END_TYPE; -- year_number
```



```
ENTITY action;
    name          : label;
    description   : OPTIONAL text;
    chosen_method : action_method;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
END_ENTITY; -- action

ENTITY action_assignment
  ABSTRACT SUPERTYPE;
    assigned_action : action;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- action_assignment

ENTITY action_directive;
    name        : label;
    description : OPTIONAL text;
    analysis    : text;
    comment     : text;
    requests    : SET [1:?] OF versioned_action_request;
END_ENTITY; -- action_directive

ENTITY action_method;
    name        : label;
    description : OPTIONAL text;
    consequence : text;
    purpose     : text;
END_ENTITY; -- action_method

ENTITY action_method_with_associated_documents
  SUBTYPE OF (action_method);
    documents : SET [1:?] OF document;
END_ENTITY; -- action_method_with_associated_documents

ENTITY action_property;
    name        : label;
    description : text;
    definition  : characterized_action_definition;
END_ENTITY; -- action_property

ENTITY action_property_representation;
    name           : label;
    description    : text;
    property       : action_property;
    representation : representation;
END_ENTITY; -- action_property_representation

ENTITY action_relationship;
    name            : label;
    description     : OPTIONAL text;
```

```
    relating_action : action;
    related_action  : action;
END_ENTITY; -- action_relationship

ENTITY action_request_assignment
  ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- action_request_assignment

ENTITY action_request_solution;
    method  : action_method;
    request : versioned_action_request;
  DERIVE
    description : text := get_description_value(SELF);
    name        : label := get_name_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- action_request_solution

ENTITY action_request_status;
    status           : label;
    assigned_request : versioned_action_request;
END_ENTITY; -- action_request_status

ENTITY action_resource;
    name        : label;
    description : OPTIONAL text;
    usage       : SET [1:?] OF supported_item;
    kind        : action_resource_type;
END_ENTITY; -- action_resource

ENTITY action_resource_relationship;
    name              : label;
    description       : OPTIONAL text;
    relating_resource : action_resource;
    related_resource  : action_resource;
END_ENTITY; -- action_resource_relationship

ENTITY action_resource_type;
    name : label;
END_ENTITY; -- action_resource_type

ENTITY action_status;
    status          : label;
    assigned_action : executed_action;
END_ENTITY; -- action_status

ENTITY address;
    internal_location       : OPTIONAL label;
    street_number           : OPTIONAL label;
    street                  : OPTIONAL label;
```



```
      postal_box              : OPTIONAL label;
      town                    : OPTIONAL label;
      region                  : OPTIONAL label;
      postal_code             : OPTIONAL label;
      country                 : OPTIONAL label;
      facsimile_number        : OPTIONAL label;
      telephone_number        : OPTIONAL label;
      electronic_mail_address : OPTIONAL label;
      telex_number            : OPTIONAL label;
    DERIVE
      name : label := get_name_value(SELF);
      url  : identifier := get_id_value(SELF);
    WHERE
      wr1: (EXISTS(internal_location) OR EXISTS(street_number) OR EXISTS(
                street) OR EXISTS(postal_box) OR EXISTS(town) OR EXISTS(
                region) OR EXISTS(postal_code) OR EXISTS(country) OR EXISTS(
                facsimile_number) OR EXISTS(telephone_number) OR EXISTS(
                electronic_mail_address) OR EXISTS(telex_number));
  END_ENTITY; -- address

  ENTITY advanced_brep_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP',
                'TECHNICAL_DATA_PACKAGING.FACETED_BREP',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
                = 1)) )) = 0);
      wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
                > 0);
      wr3: (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT ('TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN
TYPEOF(fcs))) ))
                = 0)) )) = 0)) )) = 0);
      wr4: (SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | ('TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL' IN
                TYPEOF(msb\manifold_solid_brep.outer)) )) = 0);
      wr5: (SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.BREP_WITH_VOIDS' IN TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids |
                csh\oriented_closed_shell.orientation )) = 0)) )) = 0);
      wr6: (SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT ('TECHNICAL_DATA_PACKAGING.' +
                'ADVANCED_BREP_SHAPE_REPRESENTATION'
                IN TYPEOF(mi\mapped_item.mapping_source.
                mapped_representation))) )) = 0);
  END_ENTITY; -- advanced_brep_shape_representation
```

```
  ENTITY advanced_face
    SUBTYPE OF (face_surface);
    WHERE
      wr1 : (SIZEOF(['TECHNICAL_DATA_PACKAGING.ELEMENTARY_SURFACE',
                  'TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE',
                  'TECHNICAL_DATA_PACKAGING.SWEPT_SURFACE'] * TYPEOF(
                  face_geometry)) = 1);
      wr2 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                  'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                   | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
                  edge_list | (NOT ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN
                  TYPEOF(oe\oriented_edge.edge_element))) )) = 0)) )) = 0);
      wr3 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                  'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                   | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
                  edge_list | (NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING.LINE',
                  'TECHNICAL_DATA_PACKAGING.CONIC',
                  'TECHNICAL_DATA_PACKAGING.POLYLINE',
                  'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE',
                  'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE'] * TYPEOF(oe.
                  edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
                   0);
      wr4 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                  'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                   | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
                  edge_list | (NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT'
                  IN TYPEOF(oe\edge.edge_start)) AND (
                  'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(oe\
                  edge.edge_start\vertex_point.vertex_geometry)) AND (
                  'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe\edge.
                  edge_end)) AND ('TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT'
                  IN TYPEOF(oe\edge.edge_end\vertex_point.vertex_geometry))))
))
                  = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                  'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                   | ('TECHNICAL_DATA_PACKAGING.ORIENTED_PATH' IN TYPEOF(
                  elp_fbnds.bound)) )) = 0);
      wr6 : ((NOT ('TECHNICAL_DATA_PACKAGING.SWEPT_SURFACE' IN TYPEOF(
                  face_geometry))) OR (SIZEOF([
                  'TECHNICAL_DATA_PACKAGING.LINE',
                  'TECHNICAL_DATA_PACKAGING.CONIC',
                  'TECHNICAL_DATA_PACKAGING.POLYLINE',
                  'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE'] * TYPEOF(
                  face_geometry\swept_surface.swept_curve)) = 1));
      wr7 : (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* bounds | (
                  'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN
TYPEOF(bnds.bound)) )
                   | (NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN
                  TYPEOF(vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex))
                  AND ('TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(
                  vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\
                  vertex_point.vertex_geometry)))) )) = 0);
      wr8 : (SIZEOF(QUERY ( bnd <* bounds | (NOT (SIZEOF([
                  'TECHNICAL_DATA_PACKAGING.EDGE_LOOP',
```



```
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP'] *
TYPEOF(bnd.bound))
                = 1)) )) = 0);
    wr9 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
               edge_list | (('TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN
               TYPEOF(oe\oriented_edge.edge_element\edge_curve.
               edge_geometry)) AND (NOT (SIZEOF(QUERY ( sc_ag <* oe.
               edge_element\edge_curve.edge_geometry\surface_curve.
               associated_geometry | (NOT (
               'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(sc_ag))) ))
               = 0))) )) = 0)) )) = 0);
    wr10: (((NOT ('TECHNICAL_DATA_PACKAGING.SWEPT_SURFACE' IN TYPEOF(
               face_geometry))) OR (NOT (
               'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(face_geometry
               \swept_surface.swept_curve))) OR (SIZEOF(face_geometry\
               swept_surface.swept_curve\polyline.points) >= 3)) AND (
               SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
               'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
               edge_list | (('TECHNICAL_DATA_PACKAGING.POLYLINE' IN
               TYPEOF(oe\oriented_edge.edge_element\edge_curve.
               edge_geometry)) AND (NOT (SIZEOF(oe\oriented_edge.
               edge_element\edge_curve.edge_geometry\polyline.points) >=
3)))
               )) = 0)) )) = 0));
  END_ENTITY; -- advanced_face

  ENTITY alternate_product_relationship;
      name       : label;
      definition : OPTIONAL text;
      alternate  : product;
      base       : product;
      basis      : text;
    UNIQUE
      ur1 : alternate, base;
    WHERE
      wr1: (alternate :<>: base);
  END_ENTITY; -- alternate_product_relationship

  ENTITY amount_of_substance_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.AMOUNT_OF_SUBSTANCE_UNIT' IN TYPEOF(
              SELF\measure_with_unit.unit_component));
  END_ENTITY; -- amount_of_substance_measure_with_unit

  ENTITY amount_of_substance_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
```

```
              .dimensions.amount_of_substance_exponent = 1) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
  END_ENTITY; -- amount_of_substance_unit

  ENTITY annotation_curve_occurrence
    SUBTYPE OF (annotation_occurrence);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.CURVE' IN TYPEOF(SELF\styled_item.
              item));
  END_ENTITY; -- annotation_curve_occurrence

  ENTITY annotation_occurrence
    SUPERTYPE OF (ONEOF (annotation_point_occurrence,
        annotation_curve_occurrence,annotation_text_occurrence))
    SUBTYPE OF (styled_item);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_ITEM' IN
              TYPEOF(SELF));
  END_ENTITY; -- annotation_occurrence

  ENTITY annotation_point_occurrence
    SUBTYPE OF (annotation_occurrence);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.POINT' IN TYPEOF(SELF\styled_item.
              item));
  END_ENTITY; -- annotation_point_occurrence

  ENTITY annotation_text
    SUBTYPE OF (mapped_item);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(SELF\
              mapped_item.mapping_target));
      wr2: ('TECHNICAL_DATA_PACKAGING.TEXT_STRING_REPRESENTATION' IN
              TYPEOF(SELF\mapped_item.mapping_source.
               mapped_representation));
      wr3: ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_ITEM' IN
              TYPEOF(SELF));
  END_ENTITY; -- annotation_text

  ENTITY annotation_text_character
    SUBTYPE OF (mapped_item);
      alignment : text_alignment;
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.CHARACTER_GLYPH_SYMBOL' IN TYPEOF(
              SELF\mapped_item.mapping_source.mapped_representation));
      wr2: ('TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(SELF\
              mapped_item.mapping_target));
      wr3: ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_ITEM' IN
              TYPEOF(SELF));
  END_ENTITY; -- annotation_text_character

  ENTITY annotation_text_occurrence
    SUBTYPE OF (annotation_occurrence);
    WHERE
      wr1: (SIZEOF(['TECHNICAL_DATA_PACKAGING.TEXT_LITERAL',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT_CHARACTER',
              'TECHNICAL_DATA_PACKAGING.DEFINED_CHARACTER_GLYPH',
              'TECHNICAL_DATA_PACKAGING.COMPOSITE_TEXT'] * TYPEOF(SELF\
```



```
            styled_item.item)) > 0);
END_ENTITY; -- annotation_text_occurrence

ENTITY application_context;
    application : label;
  DERIVE
    description : text := get_description_value(SELF);
    id          : identifier := get_id_value(SELF);
  INVERSE
    context_elements : SET [1:?] OF application_context_element FOR
                          frame_of_reference;
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
END_ENTITY; -- application_context

ENTITY application_context_element
  SUPERTYPE OF (ONEOF (library_context,product_concept_context,
      product_context,product_definition_context));
    name              : label;
    frame_of_reference : application_context;
END_ENTITY; -- application_context_element

ENTITY application_context_relationship;
    name            : label;
    description     : OPTIONAL text;
    relating_context : application_context;
    related_context  : application_context;
END_ENTITY; -- application_context_relationship

ENTITY application_protocol_definition;
    status                                  : label;
    application_interpreted_model_schema_name : label;
    application_protocol_year               : year_number;
    application                             : application_context;
END_ENTITY; -- application_protocol_definition

ENTITY applied_action_assignment
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF action_item;
END_ENTITY; -- applied_action_assignment

ENTITY applied_action_request_assignment
  SUBTYPE OF (action_request_assignment);
    items : SET [1:?] OF action_request_item;
END_ENTITY; -- applied_action_request_assignment

ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
    items : SET [1:?] OF approval_item;
END_ENTITY; -- applied_approval_assignment

ENTITY applied_certification_assignment
  SUBTYPE OF (certification_assignment);
    items : SET [1:?] OF certification_item;
END_ENTITY; -- applied_certification_assignment
```

```
ENTITY applied_classification_assignment
  SUBTYPE OF (classification_assignment);
    items : SET [1:?] OF classification_item;
END_ENTITY; -- applied_classification_assignment

ENTITY applied_contract_assignment
  SUBTYPE OF (contract_assignment);
    items : SET [1:?] OF contract_item;
END_ENTITY; -- applied_contract_assignment

ENTITY applied_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
    items : SET [1:?] OF date_and_time_item;
END_ENTITY; -- applied_date_and_time_assignment

ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
    items : SET [1:?] OF date_item;
END_ENTITY; -- applied_date_assignment

ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
    items : SET [1:?] OF document_reference_item;
END_ENTITY; -- applied_document_reference

ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
    items : SET [1:?] OF document_reference_item;
END_ENTITY; -- applied_document_usage_constraint_assignment

ENTITY applied_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
    items : SET [1:?] OF effectivity_item;
END_ENTITY; -- applied_effectivity_assignment

ENTITY applied_effectivity_context_assignment
  SUBTYPE OF (effectivity_context_assignment);
    items : SET [1:?] OF effectivity_context_item;
END_ENTITY; -- applied_effectivity_context_assignment

ENTITY applied_event_occurrence_assignment
  SUBTYPE OF (event_occurrence_assignment);
    items : SET [1:?] OF event_occurrence_item;
END_ENTITY; -- applied_event_occurrence_assignment

ENTITY applied_external_identification_assignment
  SUBTYPE OF (external_identification_assignment);
    items : SET [1:?] OF external_identification_item;
END_ENTITY; -- applied_external_identification_assignment

ENTITY applied_identification_assignment
  SUBTYPE OF (identification_assignment);
    items : SET [1:?] OF identification_item;
END_ENTITY; -- applied_identification_assignment

ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
    items : SET [1:?] OF organization_item;
END_ENTITY; -- applied_organization_assignment
```



```
ENTITY applied_organizational_project_assignment
  SUBTYPE OF (organizational_project_assignment);
    items : SET [1:?] OF organizational_project_item;
END_ENTITY; -- applied_organizational_project_assignment

ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
    items : SET [1:?] OF person_and_organization_item;
END_ENTITY; -- applied_person_and_organization_assignment

ENTITY applied_presented_item
  SUBTYPE OF (presented_item);
    items : SET [1:?] OF presented_item_item;
END_ENTITY; -- applied_presented_item

ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
    items : SET [1:?] OF security_classification_item;
END_ENTITY; -- applied_security_classification_assignment

ENTITY approval;
    status : approval_status;
    level  : label;
END_ENTITY; -- approval

ENTITY approval_assignment
  ABSTRACT SUPERTYPE;
    assigned_approval : approval;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- approval_assignment

ENTITY approval_date_time;
    date_time      : date_time_select;
    dated_approval : approval;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- approval_date_time

ENTITY approval_person_organization;
    person_organization : person_organization_select;
    authorized_approval : approval;
    role                : approval_role;
END_ENTITY; -- approval_person_organization

ENTITY approval_relationship;
    name              : label;
    description       : OPTIONAL text;
    relating_approval : approval;
    related_approval  : approval;
END_ENTITY; -- approval_relationship
```

```
ENTITY approval_role;
    role : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- approval_role

ENTITY approval_status;
    name : label;
END_ENTITY; -- approval_status

ENTITY area_dependent_annotation_representation
  SUBTYPE OF (presentation_representation);
  WHERE
    wr1: (SIZEOF(QUERY ( item <* SELF\representation.items | (NOT (
              SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
              'ANNOTATION_OCCURRENCE',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT'] * TYPEOF(item))
              = 1)) )) = 0);
    wr2: (SIZEOF(QUERY ( item <* SELF\representation.items | (
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_OCCURRENCE' IN TYPEOF(
              item)) )) >= 1);
END_ENTITY; -- area_dependent_annotation_representation

ENTITY area_in_set;
    area   : presentation_area;
    in_set : presentation_set;
END_ENTITY; -- area_in_set

ENTITY area_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.AREA_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
END_ENTITY; -- area_measure_with_unit

ENTITY area_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 2) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- area_unit

ENTITY assembly_component_usage
  SUPERTYPE OF (ONEOF (next_assembly_usage_occurrence,
      specified_higher_usage_occurrence,promissory_usage_occurrence))
  SUBTYPE OF (product_definition_usage);
    reference_designator : OPTIONAL identifier;
END_ENTITY; -- assembly_component_usage
```



```
  ENTITY assembly_component_usage_substitute;
      name       : label;
      definition : OPTIONAL text;
      base       : assembly_component_usage;
      substitute : assembly_component_usage;
    UNIQUE
      ur1 : base, substitute;
    WHERE
      wr1: (base.relating_product_definition :=: substitute.
               relating_product_definition);
      wr2: (base :<>: substitute);
  END_ENTITY; -- assembly_component_usage_substitute

  ENTITY assembly_component_usage_substitute_with_ranking
    SUBTYPE OF (assembly_component_usage_substitute);
      ranking           : INTEGER;
      ranking_rationale : text;
  END_ENTITY; -- assembly_component_usage_substitute_with_ranking

  ENTITY axis1_placement
    SUBTYPE OF (placement);
      axis : OPTIONAL direction;
    DERIVE
      z : direction := NVL(normalise(axis),dummy_gri ||
direction([0,0,1]));
    WHERE
      wr1: (SELF\geometric_representation_item.dim = 3);
  END_ENTITY; -- axis1_placement

  ENTITY axis2_placement_2d
    SUBTYPE OF (placement);
      ref_direction : OPTIONAL direction;
    DERIVE
      p : LIST [2:2] OF direction := build_2axes(ref_direction);
    WHERE
      wr1: (SELF\geometric_representation_item.dim = 2);
  END_ENTITY; -- axis2_placement_2d

  ENTITY axis2_placement_3d
    SUBTYPE OF (placement);
      axis          : OPTIONAL direction;
      ref_direction : OPTIONAL direction;
    DERIVE
      p : LIST [3:3] OF direction := build_axes(axis,ref_direction);
    WHERE
      wr1: (SELF\placement.location.dim = 3);
      wr2: ((NOT EXISTS(axis)) OR (axis.dim = 3));
      wr3: ((NOT EXISTS(ref_direction)) OR (ref_direction.dim = 3));
      wr4: ((NOT EXISTS(axis)) OR (NOT EXISTS(ref_direction)) OR (
              cross_product(axis,ref_direction).magnitude > 0));
  END_ENTITY; -- axis2_placement_3d

  ENTITY b_spline_curve
    SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
        quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve)
    SUBTYPE OF (bounded_curve);
      degree              : INTEGER;
      control_points_list : LIST [2:?] OF cartesian_point;
      curve_form          : b_spline_curve_form;
```

```
    closed_curve        : LOGICAL;
    self_intersect      : LOGICAL;
  DERIVE
    upper_index_on_control_points : INTEGER := SIZEOF(
                                       control_points_list) - 1;
    control_points                : ARRAY [0:
                                       upper_index_on_control_points] OF
                                        cartesian_point := list_to_array(
                                       control_points_list,0,
                                       upper_index_on_control_points);
  WHERE
    wr1: (('TECHNICAL_DATA_PACKAGING.UNIFORM_CURVE' IN TYPEOF(SELF)) OR
              ('TECHNICAL_DATA_PACKAGING.QUASI_UNIFORM_CURVE' IN TYPEOF(
              SELF)) OR ('TECHNICAL_DATA_PACKAGING.BEZIER_CURVE' IN
              TYPEOF(SELF)) OR (
              'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE_WITH_KNOTS' IN
              TYPEOF(SELF)));
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
    knot_multiplicities : LIST [2:?] OF INTEGER;
    knots               : LIST [2:?] OF parameter_value;
    knot_spec           : knot_type;
  DERIVE
    upper_index_on_knots : INTEGER := SIZEOF(knots);
  WHERE
    wr1: constraints_param_b_spline(degree,upper_index_on_knots,
              upper_index_on_control_points,knot_multiplicities,knots);
    wr2: (SIZEOF(knot_multiplicities) = upper_index_on_knots);
END_ENTITY; -- b_spline_curve_with_knots

ENTITY b_spline_surface
  SUPERTYPE OF (ONEOF (b_spline_surface_with_knots,uniform_surface,
      quasi_uniform_surface,bezier_surface) ANDOR
      rational_b_spline_surface)
  SUBTYPE OF (bounded_surface);
    u_degree            : INTEGER;
    v_degree            : INTEGER;
    control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
    surface_form        : b_spline_surface_form;
    u_closed            : LOGICAL;
    v_closed            : LOGICAL;
    self_intersect      : LOGICAL;
  DERIVE
    u_upper        : INTEGER := SIZEOF(control_points_list) - 1;
    v_upper        : INTEGER := SIZEOF(control_points_list[1]) - 1;
    control_points : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
                         cartesian_point := make_array_of_array(
                        control_points_list,0,u_upper,0,v_upper);
  WHERE
    wr1: (('TECHNICAL_DATA_PACKAGING.UNIFORM_SURFACE' IN TYPEOF(SELF))
              OR ('TECHNICAL_DATA_PACKAGING.QUASI_UNIFORM_SURFACE' IN
              TYPEOF(SELF)) OR ('TECHNICAL_DATA_PACKAGING.BEZIER_SURFACE'
              IN TYPEOF(SELF)) OR (
              'TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE_WITH_KNOTS' IN
              TYPEOF(SELF)));
END_ENTITY; -- b_spline_surface
```



```
ENTITY b_spline_surface_with_knots
  SUBTYPE OF (b_spline_surface);
    u_multiplicities : LIST [2:?] OF INTEGER;
    v_multiplicities : LIST [2:?] OF INTEGER;
    u_knots          : LIST [2:?] OF parameter_value;
    v_knots          : LIST [2:?] OF parameter_value;
    knot_spec        : knot_type;
  DERIVE
    knot_u_upper : INTEGER := SIZEOF(u_knots);
    knot_v_upper : INTEGER := SIZEOF(v_knots);
  WHERE
    wr1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,
             knot_u_upper,SELF\b_spline_surface.u_upper,u_multiplicities,
             u_knots);
    wr2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
             knot_v_upper,SELF\b_spline_surface.v_upper,v_multiplicities,
             v_knots);
    wr3: (SIZEOF(u_multiplicities) = knot_u_upper);
    wr4: (SIZEOF(v_multiplicities) = knot_v_upper);
END_ENTITY; -- b_spline_surface_with_knots

ENTITY background_colour
  SUBTYPE OF (colour);
    presentation : area_or_view;
  UNIQUE
    ur1 : presentation;
END_ENTITY; -- background_colour

ENTITY bezier_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve

ENTITY bezier_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

ENTITY block
  SUBTYPE OF (geometric_representation_item);
    position : axis2_placement_3d;
    x        : positive_length_measure;
    y        : positive_length_measure;
    z        : positive_length_measure;
END_ENTITY; -- block

ENTITY boolean_result
  SUBTYPE OF (geometric_representation_item);
    operator       : boolean_operator;
    first_operand  : boolean_operand;
    second_operand : boolean_operand;
END_ENTITY; -- boolean_result

ENTITY boundary_curve
  SUBTYPE OF (composite_curve_on_surface);
  WHERE
    wr1: SELF\composite_curve.closed_curve;
END_ENTITY; -- boundary_curve
```

```
  ENTITY bounded_curve
    SUPERTYPE OF (ONEOF (polyline,b_spline_curve,trimmed_curve,
        bounded_pcurve,bounded_surface_curve,composite_curve))
    SUBTYPE OF (curve);
  END_ENTITY; -- bounded_curve

  ENTITY bounded_pcurve
    SUBTYPE OF (pcurve, bounded_curve);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE' IN TYPEOF(SELF\pcurve
              .reference_to_curve.items[1]));
  END_ENTITY; -- bounded_pcurve

  ENTITY bounded_surface
    SUPERTYPE OF (ONEOF (b_spline_surface,rectangular_trimmed_surface,
        curve_bounded_surface,rectangular_composite_surface))
    SUBTYPE OF (surface);
  END_ENTITY; -- bounded_surface

  ENTITY bounded_surface_curve
    SUBTYPE OF (surface_curve, bounded_curve);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE' IN TYPEOF(SELF\
              surface_curve.curve_3d));
  END_ENTITY; -- bounded_surface_curve

  ENTITY box_domain;
      corner  : cartesian_point;
      xlength : positive_length_measure;
      ylength : positive_length_measure;
      zlength : positive_length_measure;
    WHERE
      wr1: (SIZEOF(QUERY ( item <* USEDIN(SELF,'') | (NOT (
              'TECHNICAL_DATA_PACKAGING.BOXED_HALF_SPACE' IN
TYPEOF(item))) ))
              = 0);
  END_ENTITY; -- box_domain

  ENTITY boxed_half_space
    SUBTYPE OF (half_space_solid);
      enclosure : box_domain;
  END_ENTITY; -- boxed_half_space

  ENTITY brep_with_voids
    SUBTYPE OF (manifold_solid_brep);
      voids : SET [1:?] OF oriented_closed_shell;
  END_ENTITY; -- brep_with_voids

  ENTITY calendar_date
    SUBTYPE OF (date);
      day_component   : day_in_month_number;
      month_component : month_in_year_number;
    WHERE
      wr1: valid_calendar_date(SELF);
  END_ENTITY; -- calendar_date

  ENTITY camera_image
    SUBTYPE OF (mapped_item);
    WHERE
```



```
    wr1: ('TECHNICAL_DATA_PACKAGING.CAMERA_USAGE' IN TYPEOF(SELF\
            mapped_item.mapping_source));
    wr2: ('TECHNICAL_DATA_PACKAGING.PLANAR_BOX' IN TYPEOF(SELF\
            mapped_item.mapping_target));
    wr3: ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_ITEM' IN
            TYPEOF(SELF));
END_ENTITY; -- camera_image

ENTITY camera_image_3d_with_scale
  SUBTYPE OF (camera_image);
  DERIVE
    scale : positive_ratio_measure := SELF\mapped_item.mapping_target\
              planar_extent.size_in_x / SELF\mapped_item.mapping_source.
              mapping_origin\camera_model_d3.perspective_of_volume.
              view_window.size_in_x;
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.CAMERA_MODEL_D3' IN TYPEOF(SELF\
            mapped_item.mapping_source.mapping_origin));
    wr2: (aspect_ratio(SELF\mapped_item.mapping_target) = aspect_ratio(
            SELF\mapped_item.mapping_source.mapping_origin\
            camera_model_d3.perspective_of_volume.view_window));
    wr3: (SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3
            .perspective_of_volume.front_plane_clipping AND SELF\
            mapped_item.mapping_source.mapping_origin\camera_model_d3.
            perspective_of_volume.view_volume_sides_clipping);
    wr4: ((SELF\mapped_item.mapping_target\planar_extent.size_in_x > 0)
            AND (SELF\mapped_item.mapping_target\planar_extent.size_in_y
            > 0));
    wr5: ((SELF\mapped_item.mapping_source.mapping_origin\
            camera_model_d3.perspective_of_volume.view_window.size_in_x
            > 0) AND (SELF\mapped_item.mapping_source.mapping_origin\
            camera_model_d3.perspective_of_volume.view_window.size_in_y
            > 0));
    wr6: ((('TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_2D') IN
            TYPEOF(SELF\mapped_item.mapping_target\
             planar_box.placement))
            AND (NOT (('TECHNICAL_DATA_PACKAGING.' +
            'AXIS2_PLACEMENT_3D') IN TYPEOF(SELF\mapped_item.
            mapping_target\planar_box.placement))));
END_ENTITY; -- camera_image_3d_with_scale

ENTITY camera_model
  SUPERTYPE OF (ONEOF (camera_model_d3))
  SUBTYPE OF (geometric_representation_item);
  WHERE
    wr1: ((SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ITEM_DEFINED_TRANSFORMATION.' + 'TRANSFORM_ITEM_1')) +
            SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'REPRESENTATION_MAP.MAPPING_ORIGIN'))) > 0);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'STYLED_ITEM.ITEM')) = 0);
END_ENTITY; -- camera_model

ENTITY camera_model_d3
  SUBTYPE OF (camera_model);
    view_reference_system : axis2_placement_3d;
    perspective_of_volume : view_volume;
  WHERE
    wr1: ((dot_product(SELF.view_reference_system.p[3],SELF.
```



```
              perspective_of_volume.view_window.placement.p[3]) = 1) AND (
              SELF.view_reference_system.location.coordinates[3] = SELF.
              perspective_of_volume.view_window.placement.location.
              coordinates[3]));
    wr2: (SELF\geometric_representation_item.dim = 3);
END_ENTITY; -- camera_model_d3

ENTITY camera_model_d3_with_hlhsr
  SUBTYPE OF (camera_model_d3);
    hidden_line_surface_removal : BOOLEAN;
END_ENTITY; -- camera_model_d3_with_hlhsr

ENTITY camera_model_with_light_sources
  SUBTYPE OF (camera_model_d3);
    sources : SET [1:?] OF light_source;
END_ENTITY; -- camera_model_with_light_sources

ENTITY camera_usage
  SUBTYPE OF (representation_map);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.PRESENTATION_REPRESENTATION' IN
              TYPEOF(SELF\representation_map.mapped_representation)));
    wr2: ('TECHNICAL_DATA_PACKAGING.CAMERA_MODEL' IN TYPEOF(SELF\
             representation_map.mapping_origin));
END_ENTITY; -- camera_usage

ENTITY cartesian_point
  SUBTYPE OF (point);
    coordinates : LIST [1:3] OF length_measure;
END_ENTITY; -- cartesian_point

ENTITY cartesian_transformation_operator
  SUPERTYPE OF (ONEOF (cartesian_transformation_operator_2d,
      cartesian_transformation_operator_3d))
  SUBTYPE OF (geometric_representation_item,
      functionally_defined_transformation);
    axis1        : OPTIONAL direction;
    axis2        : OPTIONAL direction;
    local_origin : cartesian_point;
    scale        : OPTIONAL REAL;
  DERIVE
    scl : REAL := NVL(scale,1);
  WHERE
    wr1: (scl > 0);
END_ENTITY; -- cartesian_transformation_operator

ENTITY cartesian_transformation_operator_2d
  SUBTYPE OF (cartesian_transformation_operator);
  DERIVE
    u : LIST [2:2] OF direction := base_axis(2,SELF\
           cartesian_transformation_operator.axis1,SELF\
           cartesian_transformation_operator.axis2,?);
  WHERE
    wr1: (SELF\geometric_representation_item.dim = 2);
END_ENTITY; -- cartesian_transformation_operator_2d

ENTITY cartesian_transformation_operator_3d
  SUBTYPE OF (cartesian_transformation_operator);
    axis3 : OPTIONAL direction;
```



```
  DERIVE
    u : LIST [3:3] OF direction := base_axis(3,SELF\
           cartesian_transformation_operator.axis1,SELF\
           cartesian_transformation_operator.axis2,axis3);
  WHERE
    wr1: (SELF\geometric_representation_item.dim = 3);
END_ENTITY; -- cartesian_transformation_operator_3d

ENTITY certification;
    name    : label;
    purpose : text;
    kind    : certification_type;
END_ENTITY; -- certification

ENTITY certification_assignment
  ABSTRACT SUPERTYPE;
    assigned_certification : certification;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
             'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- certification_assignment

ENTITY certification_type;
    description : label;
END_ENTITY; -- certification_type

ENTITY character_glyph_style_outline;
    outline_style : curve_style;
END_ENTITY; -- character_glyph_style_outline

ENTITY character_glyph_style_stroke;
    stroke_style : curve_style;
END_ENTITY; -- character_glyph_style_stroke

ENTITY character_glyph_symbol
  SUBTYPE OF (symbol_representation);
    character_box  : planar_extent;
    baseline_ratio : ratio_measure;
  DERIVE
    box_height : length_measure := character_box.size_in_y;
  WHERE
    wr1: ((0 <= baseline_ratio) AND (baseline_ratio <= 1));
    wr2: item_in_context(SELF.character_box,SELF\representation.
             context_of_items);
    wr3: ('TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN TYPEOF(
             SELF.box_height));
END_ENTITY; -- character_glyph_symbol

ENTITY characterized_object;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- characterized_object

ENTITY circle
  SUBTYPE OF (conic);
    radius : positive_length_measure;
END_ENTITY; -- circle
```

```
ENTITY class
  SUBTYPE OF (group);
END_ENTITY; -- class

ENTITY class_system
  SUBTYPE OF (group);
END_ENTITY; -- class_system

ENTITY classification_assignment
  ABSTRACT SUPERTYPE;
    assigned_class : group;
    role           : classification_role;
END_ENTITY; -- classification_assignment

ENTITY classification_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- classification_role

ENTITY closed_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell

ENTITY colour;
END_ENTITY; -- colour

ENTITY colour_rgb
  SUBTYPE OF (colour_specification);
    red   : REAL;
    green : REAL;
    blue  : REAL;
  WHERE
    wr1: ((0 <= red) AND (red <= 1));
    wr2: ((0 <= green) AND (green <= 1));
    wr3: ((0 <= blue) AND (blue <= 1));
END_ENTITY; -- colour_rgb

ENTITY colour_specification
  SUBTYPE OF (colour);
    name : label;
END_ENTITY; -- colour_specification

ENTITY composite_curve
  SUBTYPE OF (bounded_curve);
    segments       : LIST [1:?] OF composite_curve_segment;
    self_intersect : LOGICAL;
  DERIVE
    n_segments   : INTEGER := SIZEOF(segments);
    closed_curve : LOGICAL := segments[n_segments].transition <>
                      discontinuous;
  WHERE
    wr1: (((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
             temp.transition = discontinuous) )) = 1)) OR (closed_curve
             AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
             discontinuous) )) = 0)));
END_ENTITY; -- composite_curve
```



```
ENTITY composite_curve_on_surface
  SUPERTYPE OF (boundary_curve)
  SUBTYPE OF (composite_curve);
  DERIVE
    basis_surface : SET [0:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: (SIZEOF(basis_surface) > 0);
    wr2: constraints_composite_curve_on_surface(SELF);
END_ENTITY; -- composite_curve_on_surface

ENTITY composite_curve_segment
  SUBTYPE OF (founded_item);
    transition   : transition_code;
    same_sense   : BOOLEAN;
    parent_curve : curve;
  INVERSE
    using_curves : BAG [1:?] OF composite_curve FOR segments;
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE' IN TYPEOF(
              parent_curve));
END_ENTITY; -- composite_curve_segment

ENTITY composite_text
  SUBTYPE OF (geometric_representation_item);
    collected_text : SET [2:?] OF text_or_character;
  WHERE
    wr1: acyclic_composite_text(SELF,SELF.collected_text);
END_ENTITY; -- composite_text

ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
    item_element : compound_item_definition;
END_ENTITY; -- compound_representation_item

ENTITY configuration_design;
    configuration : configuration_item;
    design        : configuration_design_item;
  DERIVE
    name        : label := get_name_value(SELF);
    description : text := get_description_value(SELF);
  UNIQUE
    ur1 : configuration, design;
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- configuration_design

ENTITY configuration_effectivity
  SUBTYPE OF (product_definition_effectivity);
    configuration : configuration_design;
  UNIQUE
    ur1 : configuration, usage, id;
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_USAGE' IN TYPEOF(
              SELF\product_definition_effectivity.usage));
END_ENTITY; -- configuration_effectivity
```

```
ENTITY configuration_item;
    id          : identifier;
    name        : label;
    description  : OPTIONAL text;
    item_concept : product_concept;
    purpose      : OPTIONAL label;
END_ENTITY; -- configuration_item

ENTITY conic
  SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
  SUBTYPE OF (curve);
    position : axis2_placement;
END_ENTITY; -- conic

ENTITY conical_surface
  SUBTYPE OF (elementary_surface);
    radius     : length_measure;
    semi_angle : plane_angle_measure;
  WHERE
    wr1: (radius >= 0);
END_ENTITY; -- conical_surface

ENTITY connected_edge_set
  SUBTYPE OF (topological_representation_item);
    ces_edges : SET [1:?] OF edge;
END_ENTITY; -- connected_edge_set

ENTITY connected_face_set
  SUPERTYPE OF (ONEOF (closed_shell,open_shell))
  SUBTYPE OF (topological_representation_item);
    cfs_faces : SET [1:?] OF face;
END_ENTITY; -- connected_face_set

ENTITY context_dependent_shape_representation;
    representation_relation      : shape_representation_relationship;
    represented_product_relation : product_definition_shape;
  DERIVE
    description : text := get_description_value(SELF);
    name        : label := get_name_value(SELF);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_RELATIONSHIP' IN
              TYPEOF(SELF.represented_product_relation.definition));
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
    wr3: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- context_dependent_shape_representation

ENTITY context_dependent_unit
  SUBTYPE OF (named_unit);
    name : label;
END_ENTITY; -- context_dependent_unit

ENTITY contract;
    name    : label;
    purpose : text;
    kind    : contract_type;
END_ENTITY; -- contract
```



```
ENTITY contract_assignment
  ABSTRACT SUPERTYPE;
    assigned_contract : contract;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- contract_assignment

ENTITY contract_relationship;
    id               : identifier;
    name             : label;
    description      : OPTIONAL text;
    relating_contract : contract;
    related_contract  : contract;
END_ENTITY; -- contract_relationship

ENTITY contract_type;
    description : label;
END_ENTITY; -- contract_type

ENTITY conversion_based_unit
  SUBTYPE OF (named_unit);
    name             : label;
    conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit

ENTITY coordinated_universal_time_offset;
    hour_offset   : INTEGER;
    minute_offset : OPTIONAL INTEGER;
    sense         : ahead_or_behind;
  DERIVE
    actual_minute_offset : INTEGER := NVL(minute_offset,0);
  WHERE
    wr1: ((0 <= hour_offset) AND (hour_offset < 24));
    wr2: ((0 <= actual_minute_offset) AND (actual_minute_offset <= 59));
    wr3: (NOT (((hour_offset <> 0) OR (actual_minute_offset <> 0)) AND (
              sense = exact)));
END_ENTITY; -- coordinated_universal_time_offset

ENTITY csg_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: (SELF.context_of_items\geometric_representation_context.
              coordinate_space_dimension = 3);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.CSG_SOLID',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D',
              'TECHNICAL_DATA_PACKAGING.SOLID_REPLICA',
              'TECHNICAL_DATA_PACKAGING.REVOLVED_FACE_SOLID',
              'TECHNICAL_DATA_PACKAGING.EXTRUDED_FACE_SOLID'] *
                TYPEOF(it))
              <> 1) )) = 0);
    wr3: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
              NOT ('TECHNICAL_DATA_PACKAGING.CSG_SHAPE_REPRESENTATION' IN
              TYPEOF(mi\mapped_item.mapping_source
```

```
                .mapped_representation))) ))
                = 0);
      wr4: (SIZEOF(QUERY ( sr <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SOLID_REPLICA' IN TYPEOF(it)) ) |
                (SIZEOF(['TECHNICAL_DATA_PACKAGING.CSG_SOLID',
                'TECHNICAL_DATA_PACKAGING.REVOLVED_FACE_SOLID',
                'TECHNICAL_DATA_PACKAGING.EXTRUDED_FACE_SOLID'] * TYPEOF(sr\
                solid_replica.parent_solid)) = 0) )) = 0);
      wr5: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D' IN
TYPEOF(it))) ))
                > 0);
  END_ENTITY; -- csg_shape_representation

  ENTITY csg_solid
    SUBTYPE OF (solid_model);
      tree_root_expression : csg_select;
  END_ENTITY; -- csg_solid

  ENTITY curve
    SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve,offset_curve_2d,
        offset_curve_3d,curve_replica))
    SUBTYPE OF (geometric_representation_item);
  END_ENTITY; -- curve

  ENTITY curve_bounded_surface
    SUBTYPE OF (bounded_surface);
      basis_surface  : surface;
      boundaries     : SET [1:?] OF boundary_curve;
      implicit_outer : BOOLEAN;
    WHERE
      wr1: ((NOT implicit_outer) OR (SIZEOF(QUERY ( temp <* boundaries | (
                'TECHNICAL_DATA_PACKAGING.OUTER_BOUNDARY_CURVE' IN TYPEOF(
                temp)) )) = 0));
      wr2: ((NOT implicit_outer) OR (
                'TECHNICAL_DATA_PACKAGING.BOUNDED_SURFACE' IN TYPEOF(
                basis_surface)));
      wr3: (SIZEOF(QUERY ( temp <* boundaries | (
                'TECHNICAL_DATA_PACKAGING.OUTER_BOUNDARY_CURVE' IN TYPEOF(
                temp)) )) <= 1);
      wr4: (SIZEOF(QUERY ( temp <* boundaries | (temp\
                composite_curve_on_surface.basis_surface[1] <> SELF.
                basis_surface) )) = 0);
  END_ENTITY; -- curve_bounded_surface

  ENTITY curve_replica
    SUBTYPE OF (curve);
      parent_curve   : curve;
      transformation : cartesian_transformation_operator;
    WHERE
      wr1: (transformation.dim = parent_curve.dim);
      wr2: acyclic_curve_replica(SELF,parent_curve);
  END_ENTITY; -- curve_replica
```



```
ENTITY curve_style;
    name         : label;
    curve_font   : curve_font_or_scaled_curve_font_select;
    curve_width  : size_select;
    curve_colour : colour;
END_ENTITY; -- curve_style

ENTITY curve_style_font;
    name         : label;
    pattern_list : LIST [1:?] OF curve_style_font_pattern;
END_ENTITY; -- curve_style_font

ENTITY curve_style_font_pattern;
    visible_segment_length   : positive_length_measure;
    invisible_segment_length : positive_length_measure;
END_ENTITY; -- curve_style_font_pattern

ENTITY curve_style_rendering;
    rendering_method     : shading_curve_method;
    rendering_properties : surface_rendering_properties;
END_ENTITY; -- curve_style_rendering

ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

ENTITY date
  SUPERTYPE OF (ONEOF (calendar_date));
    year_component : year_number;
END_ENTITY; -- date

ENTITY date_and_time;
    date_component : date;
    time_component : local_time;
END_ENTITY; -- date_and_time

ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
    assigned_date_and_time : date_and_time;
    role                   : date_time_role;
END_ENTITY; -- date_and_time_assignment

ENTITY date_assignment
  ABSTRACT SUPERTYPE;
    assigned_date : date;
    role          : date_role;
END_ENTITY; -- date_assignment

ENTITY date_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- date_role
```

```
ENTITY date_time_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- date_time_role

ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
    effectivity_end_date   : OPTIONAL date_time_or_event_occurrence;
    effectivity_start_date : date_time_or_event_occurrence;
END_ENTITY; -- dated_effectivity

ENTITY defined_character_glyph
  SUBTYPE OF (geometric_representation_item);
    definition : defined_glyph_select;
    placement  : axis2_placement;
END_ENTITY; -- defined_character_glyph

ENTITY definitional_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.PARAMETRIC_REPRESENTATION_CONTEXT'
              IN TYPEOF(SELF\representation.context_of_items));
END_ENTITY; -- definitional_representation

ENTITY degenerate_pcurve
  SUBTYPE OF (point);
    basis_surface      : surface;
    reference_to_curve : definitional_representation;
  WHERE
    wr1: (SIZEOF(reference_to_curve\representation.items) = 1);
    wr2: ('TECHNICAL_DATA_PACKAGING.CURVE' IN TYPEOF(reference_to_curve\
              representation.items[1]));
    wr3: (reference_to_curve\representation.items[1]\
              geometric_representation_item.dim = 2);
END_ENTITY; -- degenerate_pcurve

ENTITY degenerate_toroidal_surface
  SUBTYPE OF (toroidal_surface);
    select_outer : BOOLEAN;
  WHERE
    wr1: (major_radius < minor_radius);
END_ENTITY; -- degenerate_toroidal_surface

ENTITY derived_unit;
    elements : SET [1:?] OF derived_unit_element;
  DERIVE
    name : label := get_name_value(SELF);
  WHERE
    wr1: ((SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (
              elements[1].exponent <> 1)));
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- derived_unit
```



```
ENTITY derived_unit_element;
    unit     : named_unit;
    exponent : REAL;
END_ENTITY; -- derived_unit_element

ENTITY description_attribute;
    attribute_value : text;
    described_item  : description_attribute_select;
END_ENTITY; -- description_attribute

ENTITY descriptive_representation_item
  SUBTYPE OF (representation_item);
    description : text;
END_ENTITY; -- descriptive_representation_item

ENTITY design_make_from_relationship
  SUBTYPE OF (product_definition_relationship);
END_ENTITY; -- design_make_from_relationship

ENTITY dimensional_exponents;
    length_exponent                   : REAL;
    mass_exponent                     : REAL;
    time_exponent                     : REAL;
    electric_current_exponent         : REAL;
    thermodynamic_temperature_exponent : REAL;
    amount_of_substance_exponent      : REAL;
    luminous_intensity_exponent       : REAL;
END_ENTITY; -- dimensional_exponents

ENTITY directed_action
  SUBTYPE OF (executed_action);
    directive : action_directive;
END_ENTITY; -- directed_action

ENTITY direction
  SUBTYPE OF (geometric_representation_item);
    direction_ratios : LIST [2:3] OF REAL;
  WHERE
    wr1: (SIZEOF(QUERY ( tmp <* direction_ratios | (tmp <> 0) )) > 0);
END_ENTITY; -- direction

ENTITY document;
    id          : identifier;
    name        : label;
    description : OPTIONAL text;
    kind        : document_type;
  INVERSE
    representation_types : SET [0:?] OF document_representation_type FOR
                              represented_document;
END_ENTITY; -- document

ENTITY document_file
  SUBTYPE OF (document, characterized_object);
  WHERE
    wr1: (SELF\characterized_object.name = '');
    wr2: (NOT EXISTS(SELF\characterized_object.description));
    wr3: (SIZEOF(QUERY ( drt <* SELF\document.representation_types | (
              drt.name IN ['digital','physical']) )) = 1);
END_ENTITY; -- document_file
```

```
ENTITY document_product_association;
    name              : label;
    description       : OPTIONAL text;
    relating_document : document;
    related_product   : product_or_formation_or_definition;
END_ENTITY; -- document_product_association

ENTITY document_product_equivalence
  SUBTYPE OF (document_product_association);
  WHERE
    wr1: (SELF.name = 'equivalence');
    wr2: ((NOT ('TECHNICAL_DATA_PACKAGING.PRODUCT' IN TYPEOF(SELF.
              related_product))) OR ((SELF.relating_document.kind.
              product_data_type = 'configuration controlled document') AND
               (SIZEOF(QUERY ( prpc <* USEDIN(SELF.related_product,
              'TECHNICAL_DATA_PACKAGING.' +
              'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc.name =
              'document') )) = 1)));
    wr3: ((NOT ('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION'
              IN TYPEOF(SELF.related_product))) OR ((SELF.
              relating_document.kind.product_data_type =
              'configuration controlled document version') AND (SIZEOF(
              QUERY ( prpc <* USEDIN(SELF.related_product\
              product_definition_formation.of_product,
              'TECHNICAL_DATA_PACKAGING.' +
              'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')
               | (prpc.name = 'document') )) = 1)));
    wr4: ((NOT ('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION' IN TYPEOF(
              SELF.related_product))) OR ((SELF.relating_document.kind.
              product_data_type =
              'configuration controlled document definition') AND (SIZEOF(
              QUERY ( prpc <* USEDIN(SELF.related_product\
              product_definition.formation.of_product,
              'TECHNICAL_DATA_PACKAGING.' +
              'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')
               | (prpc.name = 'document') )) = 1)));
END_ENTITY; -- document_product_equivalence

ENTITY document_reference
  ABSTRACT SUPERTYPE;
    assigned_document : document;
    source            : label;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- document_reference

ENTITY document_relationship;
    name              : label;
    description       : OPTIONAL text;
    relating_document : document;
    related_document  : document;
END_ENTITY; -- document_relationship
```



```
  ENTITY document_representation_type;
      name                : label;
      represented_document : document;
  END_ENTITY; -- document_representation_type

  ENTITY document_type;
      product_data_type : label;
  END_ENTITY; -- document_type

  ENTITY document_usage_constraint;
      source               : document;
      subject_element      : label;
      subject_element_value : text;
  END_ENTITY; -- document_usage_constraint

  ENTITY document_usage_constraint_assignment
    ABSTRACT SUPERTYPE;
      assigned_document_usage : document_usage_constraint;
      role                    : document_usage_role;
  END_ENTITY; -- document_usage_constraint_assignment

  ENTITY document_usage_role;
      name        : label;
      description : OPTIONAL text;
  END_ENTITY; -- document_usage_role

  ENTITY draughting_approval_assignment
    SUBTYPE OF (approval_assignment);
      approved_items : SET [1:?] OF approved_item;
    WHERE
      wr1: (SIZEOF(QUERY ( item <* approved_items | (
                'TECHNICAL_DATA_PACKAGING.DRAWING_REVISION' IN TYPEOF(item))
))
               <= 1);
  END_ENTITY; -- draughting_approval_assignment

  ENTITY draughting_callout
    SUBTYPE OF (geometric_representation_item);
      contents : SET [1:?] OF draughting_callout_element;
  END_ENTITY; -- draughting_callout

  ENTITY draughting_drawing_revision
    SUBTYPE OF (drawing_revision);
    WHERE
      wr1 : ((SIZEOF(USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET')) >= 1) AND (
                SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                'TECHNICAL_DATA_PACKAGING.DRAWING_SHEET_REVISION' IN
                TYPEOF(ais.area))) )) = 0));
      wr2 : (SIZEOF(QUERY ( app_ass <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_APPROVAL_ASSIGNMENT.APPROVED_ITEMS') | (NOT (
                SIZEOF(USEDIN(app_ass.assigned_approval,
                'TECHNICAL_DATA_PACKAGING.' +
                'APPROVAL_DATE_TIME.DATED_APPROVAL')) = 1)) )) = 0);
      wr3 : (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( app_ass <* USEDIN(ais.area,
```

```
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_APPROVAL_ASSIGNMENT.APPROVED_ITEMS') | (NOT (
                SIZEOF(USEDIN(app_ass.assigned_approval,
                'TECHNICAL_DATA_PACKAGING.' +
                'APPROVAL_DATE_TIME.DATED_APPROVAL')) = 1)) )) = 0)) )) =
0);
      wr4 : (SIZEOF(QUERY ( app_ass <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_APPROVAL_ASSIGNMENT.APPROVED_ITEMS') | (NOT (
                SIZEOF(USEDIN(app_ass.assigned_approval,
                'TECHNICAL_DATA_PACKAGING.' +
                'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL')) >= 1))
))
                = 0);
      wr5 : (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( app_ass <* USEDIN(ais.area,
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_APPROVAL_ASSIGNMENT.APPROVED_ITEMS') | (NOT (
                SIZEOF(USEDIN(app_ass.assigned_approval,
                'TECHNICAL_DATA_PACKAGING.' +
                'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL')) >= 1))
))
                = 0)) )) = 0);
      wr6 : (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_TITLE.ITEMS')) <= 1);
      wr7 : (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(USEDIN(ais.area,'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_TITLE.ITEMS')) <= 1)) )) = 0);
      wr8 : (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( item <* ais.area.items | (NOT (SIZEOF(
                TYPEOF(item) * ['TECHNICAL_DATA_PACKAGING.STYLED_ITEM',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT',
                'TECHNICAL_DATA_PACKAGING.PLANAR_BOX']) = 1)) )) = 0)) )) =
                 0);
      wr9 : (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( item <* ais.area.items |
(SIZEOF(TYPEOF(item)
                * ['TECHNICAL_DATA_PACKAGING.STYLED_ITEM',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM']) = 1) )) > 0)) )) =
                 0);
      wr10: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( p_b <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.PLANAR_BOX' IN TYPEOF(item)) ) |
                (NOT (SIZEOF(USEDIN(p_b,'TECHNICAL_DATA_PACKAGING' +
                '.PRESENTATION_SIZE.SIZE')) = 1)) )) = 0)) ))
                = 0);
      wr11: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN
                TYPEOF(mi.mapping_source.mapped_representation))) )) = 0))
))
```



```
                = 0);
      wr12: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( a2p <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(item))
)
                 | (NOT (SIZEOF(USEDIN(a2p,'TECHNICAL_DATA_PACKAGING.' +
                'MAPPED_ITEM.MAPPING_TARGET')) > 0)) )) = 0)) )) = 0);
      wr13: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(ais.area.context_of_items
                .representations_in_context)
                = 1)) )) = 0);
      wr14: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT (SIZEOF(USEDIN(mi.mapping_source.
                mapped_representation,'TECHNICAL_DATA_PACKAGING.' +
                'REPRESENTATION_MAP.MAPPED_REPRESENTATION')) = 1)) )) = 0))
))
                = 0);
      wr15: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT (SIZEOF(QUERY ( pv_item <* mi.mapping_source.
                mapped_representation.items | (NOT (SIZEOF(TYPEOF(pv_item)
                * ['TECHNICAL_DATA_PACKAGING.' + 'STYLED_ITEM',
                'TECHNICAL_DATA_PACKAGING.CAMERA_IMAGE',
                'TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT']) = 1)) ))
                = 0)) )) = 0)) )) = 0);
      wr16: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT (SIZEOF(QUERY ( pv_item <* mi.mapping_source.
                mapped_representation.items | (
                'TECHNICAL_DATA_PACKAGING.CAMERA_IMAGE' IN TYPEOF(pv_item))
))
                = 1)) )) = 0)) )) = 0);
      wr17: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT (SIZEOF(QUERY ( a2p <* QUERY ( pv_item <* mi.
                mapping_source.mapped_representation.items | (
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(
                pv_item)) ) | (('TECHNICAL_DATA_PACKAGING.' +
                'REPRESENTATION_MAP.MAP_USAGE') IN ROLESOF(a2p)) )) =
                1)) )) = 0)) )) = 0);
      wr18: (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET.IN_SET') | (NOT (
                SIZEOF(QUERY ( mi <* QUERY ( item <* ais.area.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) )
                 | (NOT (SIZEOF(mi.mapping_source.mapped_representation.
```

```
                context_of_items.representations_in_context) = 1)) )) = 0))
))
                = 0);
  END_ENTITY; -- draughting_drawing_revision

  ENTITY draughting_pre_defined_colour
    SUBTYPE OF (pre_defined_colour);
    WHERE
      wr1: (SELF.name IN ['red','green','blue','yellow','magenta','cyan',
              'black','white']);
  END_ENTITY; -- draughting_pre_defined_colour

  ENTITY draughting_pre_defined_curve_font
    SUBTYPE OF (pre_defined_curve_font);
    WHERE
      wr1: (SELF.name IN ['continuous','chain','chain double dash',
              'dashed','dotted']);
  END_ENTITY; -- draughting_pre_defined_curve_font

  ENTITY draughting_title;
      items    : SET [1:?] OF draughting_titled_item;
      language : label;
      contents : text;
  END_ENTITY; -- draughting_title

  ENTITY drawing_definition;
      drawing_number : identifier;
      drawing_type   : OPTIONAL label;
  END_ENTITY; -- drawing_definition

  ENTITY drawing_revision
    SUBTYPE OF (presentation_set);
      revision_identifier : identifier;
      drawing_identifier  : drawing_definition;
      intended_scale      : OPTIONAL text;
    UNIQUE
      ur1 : revision_identifier, drawing_identifier;
  END_ENTITY; -- drawing_revision

  ENTITY drawing_sheet_revision
    SUBTYPE OF (presentation_area);
      revision_identifier : identifier;
    WHERE
      wr1: (SIZEOF(QUERY ( item <* SELF\representation.items | ((
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) AND
              ('TECHNICAL_DATA_PACKAGING.DRAWING_SHEET_REVISION' IN
              TYPEOF(item\mapped_item.mapping_source
               .mapped_representation)))
               ))  = 0);
  END_ENTITY; -- drawing_sheet_revision

  ENTITY edge
    SUPERTYPE OF (ONEOF (edge_curve,oriented_edge))
    SUBTYPE OF (topological_representation_item);
      edge_start : vertex;
      edge_end   : vertex;
  END_ENTITY; -- edge
```



```
  ENTITY edge_based_wireframe_model
    SUBTYPE OF (geometric_representation_item);
      ebwm_boundary : SET [1:?] OF connected_edge_set;
  END_ENTITY; -- edge_based_wireframe_model

  ENTITY edge_based_wireframe_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL',
               'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
               'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
               = 1)) )) = 0);
      wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL',
               'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
               >= 1);
      wr3: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
               edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
               QUERY ( edges <* eb.ces_edges | (NOT (
               'TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN TYPEOF(edges))) ))
               = 0)) )) = 0)) )) = 0);
      wr4: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
               edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
               QUERY ( pline_edges <* QUERY ( edges <* eb.ces_edges | (
               'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(edges\
               edge_curve.edge_geometry)) ) | (NOT (SIZEOF(pline_edges\
               edge_curve.edge_geometry\polyline.points) > 2)) )) = 0)) ))
               = 0)) )) = 0);
      wr5: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
               edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
               QUERY ( edges <* eb.ces_edges | (NOT ((
               'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(edges.
               edge_start)) AND ('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN
                TYPEOF(edges.edge_end)))) )) = 0)) )) = 0)) )) = 0);
      wr6: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
               edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
               QUERY ( edges <* eb.ces_edges | (NOT
               valid_wireframe_edge_curve(edges\edge_curve.edge_geometry))
))
               = 0)) )) = 0)) )) = 0);
      wr7: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.EDGE_BASED_WIREFRAME_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
               edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
               QUERY ( edges <* eb.ces_edges | (NOT (
               valid_wireframe_vertex_point(edges.edge_start\vertex_point.
               vertex_geometry) AND valid_wireframe_vertex_point(edges.
               edge_end\vertex_point.vertex_geometry))) )) = 0)) )) = 0))
))
```

```
            = 0);
    wr8: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
            NOT (('TECHNICAL_DATA_PACKAGING.' +
            'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
            mapped_item.mapping_source.mapped_representation))) )) = 0);
    wr9: (SELF.context_of_items\geometric_representation_context.
            coordinate_space_dimension = 3);
END_ENTITY; -- edge_based_wireframe_shape_representation

ENTITY edge_curve
  SUBTYPE OF (edge, geometric_representation_item);
    edge_geometry : curve;
    same_sense    : BOOLEAN;
END_ENTITY; -- edge_curve

ENTITY edge_loop
  SUBTYPE OF (loop, path);
  DERIVE
    ne : INTEGER := SIZEOF(SELF\path.edge_list);
  WHERE
    wr1: (SELF\path.edge_list[1].edge_start :=: SELF\path.edge_list[ne].
            edge_end);
END_ENTITY; -- edge_loop

ENTITY effectivity
  SUPERTYPE OF (ONEOF (serial_numbered_effectivity,dated_effectivity,
      lot_effectivity,time_interval_based_effectivity));
    id : identifier;
  DERIVE
    name        : label := get_name_value(SELF);
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- effectivity

ENTITY effectivity_assignment
  ABSTRACT SUPERTYPE;
    assigned_effectivity : effectivity;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- effectivity_assignment

ENTITY effectivity_context_assignment
  ABSTRACT SUPERTYPE;
    assigned_effectivity_assignment : effectivity_assignment;
    role                            : effectivity_context_role;
END_ENTITY; -- effectivity_context_assignment

ENTITY effectivity_context_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- effectivity_context_role
```



```
  ENTITY effectivity_relationship;
      name                 : label;
      description          : OPTIONAL text;
      related_effectivity  : effectivity;
      relating_effectivity : effectivity;
  END_ENTITY; -- effectivity_relationship

  ENTITY electric_current_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.ELECTRIC_CURRENT_UNIT' IN TYPEOF(
              SELF\measure_with_unit.unit_component));
  END_ENTITY; -- electric_current_measure_with_unit

  ENTITY electric_current_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 1) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
  END_ENTITY; -- electric_current_unit

  ENTITY elementary_brep_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP',
              'TECHNICAL_DATA_PACKAGING.FACETED_BREP',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] *
TYPEOF(it))
              = 1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
              > 0);
      wr3 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
               | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
              SIZEOF(QUERY ( fcs <* csh.cfs_faces | (NOT (
              'TECHNICAL_DATA_PACKAGING.FACE_SURFACE' IN TYPEOF(fcs))) ))
              = 0)) )) = 0)) )) = 0);
      wr4 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
               | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
              SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
              NOT ('TECHNICAL_DATA_PACKAGING.ELEMENTARY_SURFACE' IN
              TYPEOF(fcs\face_surface.face_geometry))) )) = 0)) )) = 0))
))
              = 0);
      wr5 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
```

```
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds
                 | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                .edge_list | (NOT ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN
                 TYPEOF(oe.edge_element))) )) = 0)) )) = 0)) )) = 0)) )) =
                0)) )) = 0);
      wr6 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds
                 | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                .edge_list | (NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING.LINE',
                'TECHNICAL_DATA_PACKAGING.CONIC',
                'TECHNICAL_DATA_PACKAGING.POLYLINE'] * TYPEOF(oe.
                edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
                 0)) )) = 0)) )) = 0)) )) = 0);
      wr7 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds
                 | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                .edge_list | (NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT'
                IN TYPEOF(oe.edge_start)) AND (
                'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe.
                edge_end)))) )) = 0)) )) = 0)) )) = 0)) )) = 0)) )) = 0);
      wr8 : (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fcs.bounds
                 | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                .edge_list | (('TECHNICAL_DATA_PACKAGING.POLYLINE' IN
                TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND (NOT
                (SIZEOF(oe\oriented_edge.edge_element\edge_curve.
                edge_geometry\polyline.points) >= 3))) )) = 0)) )) = 0)) ))
                = 0)) )) = 0)) )) = 0);
      wr9 : (SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | ('TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL' IN
                TYPEOF(msb\manifold_solid_brep.outer)) )) = 0);
      wr10: (SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.BREP_WITH_VOIDS' IN TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids |
                csh\oriented_closed_shell.orientation )) = 0)) )) = 0);
      wr11: (SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) |
```



```
                (NOT('TECHNICAL_DATA_PACKAGING.' +
                 'ELEMENTARY_BREP_SHAPE_REPRESENTATION'
                IN TYPEOF(mi\mapped_item.mapping_source.
                mapped_representation))) )) = 0);
      wr12: (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (
                SIZEOF(QUERY ( fcs <* csh\connected_face_set.cfs_faces | (
                NOT (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fcs.bounds
                 | ('TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT'
                IN TYPEOF(vlp_fbnds\face_bound.bound\vertex_loop.
                loop_vertex)) AND (
                'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(
                vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\
                vertex_point.vertex_geometry)))) )) = 0)) )) = 0)) )) = 0))
))
                = 0);
  END_ENTITY; -- elementary_brep_shape_representation

  ENTITY elementary_surface
    SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
        spherical_surface,toroidal_surface))
    SUBTYPE OF (surface);
      position : axis2_placement_3d;
  END_ENTITY; -- elementary_surface

  ENTITY ellipse
    SUBTYPE OF (conic);
      semi_axis_1 : positive_length_measure;
      semi_axis_2 : positive_length_measure;
  END_ENTITY; -- ellipse

  ENTITY evaluated_degenerate_pcurve
    SUBTYPE OF (degenerate_pcurve);
      equivalent_point : cartesian_point;
  END_ENTITY; -- evaluated_degenerate_pcurve

  ENTITY event_occurrence;
      id          : identifier;
      name        : label;
      description : OPTIONAL text;
  END_ENTITY; -- event_occurrence

  ENTITY event_occurrence_assignment
    ABSTRACT SUPERTYPE;
      assigned_event_occurrence : event_occurrence;
      role                      : event_occurrence_role;
  END_ENTITY; -- event_occurrence_assignment

  ENTITY event_occurrence_relationship;
      name           : label;
      description    : OPTIONAL text;
      relating_event : event_occurrence;
      related_event  : event_occurrence;
  END_ENTITY; -- event_occurrence_relationship
```

```
ENTITY event_occurrence_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- event_occurrence_role

ENTITY executed_action
  SUBTYPE OF (action);
END_ENTITY; -- executed_action

ENTITY external_identification_assignment
  ABSTRACT SUPERTYPE
  SUBTYPE OF (identification_assignment);
    source : external_source;
END_ENTITY; -- external_identification_assignment

ENTITY external_source;
    source_id : source_item;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- external_source

ENTITY externally_defined_character_glyph
  SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_character_glyph

ENTITY externally_defined_class
  SUBTYPE OF (externally_defined_item, class);
END_ENTITY; -- externally_defined_class

ENTITY externally_defined_general_property
  SUBTYPE OF (externally_defined_item, general_property);
END_ENTITY; -- externally_defined_general_property

ENTITY externally_defined_item;
    item_id : source_item;
    source  : external_source;
END_ENTITY; -- externally_defined_item

ENTITY externally_defined_planar_box
  SUBTYPE OF (externally_defined_item, planar_box);
END_ENTITY; -- externally_defined_planar_box

ENTITY externally_defined_style
  SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_style

ENTITY externally_defined_symbol
  SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_symbol

ENTITY externally_defined_symbol_and_placement
  SUBTYPE OF (externally_defined_symbol, placement);
END_ENTITY; -- externally_defined_symbol_and_placement
```



```
  ENTITY externally_defined_text_font
    SUBTYPE OF (externally_defined_item);
  END_ENTITY; -- externally_defined_text_font

  ENTITY extruded_face_solid
    SUBTYPE OF (swept_face_solid);
      extruded_direction : direction;
      depth              : positive_length_measure;
    WHERE
      wr1: (dot_product(SELF\swept_face_solid.swept_face.face_geometry\
              elementary_surface.position.p[3],extruded_direction) <> 0);
  END_ENTITY; -- extruded_face_solid

  ENTITY face
    SUPERTYPE OF (ONEOF (face_surface,oriented_face))
    SUBTYPE OF (topological_representation_item);
      bounds : SET [1:?] OF face_bound;
    WHERE
      wr1: (NOT mixed_loop_type_set(list_to_set(list_face_loops(SELF))));
      wr2: (SIZEOF(QUERY ( temp <* bounds | (
              'TECHNICAL_DATA_PACKAGING.FACE_OUTER_BOUND' IN TYPEOF(temp))
))
              <= 1);
  END_ENTITY; -- face

  ENTITY face_based_surface_model
    SUBTYPE OF (geometric_representation_item);
      fbsm_faces : SET [1:?] OF connected_face_set;
  END_ENTITY; -- face_based_surface_model

  ENTITY face_bound
    SUBTYPE OF (topological_representation_item);
      bound       : loop;
      orientation : BOOLEAN;
  END_ENTITY; -- face_bound

  ENTITY face_outer_bound
    SUBTYPE OF (face_bound);
  END_ENTITY; -- face_outer_bound

  ENTITY face_surface
    SUBTYPE OF (face, geometric_representation_item);
      face_geometry : surface;
      same_sense    : BOOLEAN;
    WHERE
      wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_SURFACE' IN TYPEOF(
              face_geometry)));
  END_ENTITY; -- face_surface

  ENTITY faceted_brep
    SUBTYPE OF (manifold_solid_brep);
  END_ENTITY; -- faceted_brep

  ENTITY faceted_brep_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SIZEOF(QUERY ( it <* items | (NOT (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.FACETED_BREP',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
```

```
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
                = 1)) )) = 0);
      wr2: (SIZEOF(QUERY ( it <* items | (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.FACETED_BREP',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
                > 0);
      wr3: (SIZEOF(QUERY ( fbrep <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.FACETED_BREP' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( csh <* msb_shells(fbrep) | (NOT (SIZEOF(
                QUERY ( fcs <* csh\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.FACE_SURFACE' IN TYPEOF(fcs)) AND
                ('TECHNICAL_DATA_PACKAGING.PLANE' IN TYPEOF(fcs\face_surface
                .face_geometry)) AND (
                'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(fcs\
                face_surface.face_geometry\elementary_surface.position.
                location)))) )) = 0)) )) = 0)) )) = 0);
      wr4: (SIZEOF(QUERY ( fbrep <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.FACETED_BREP' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( csh <* msb_shells(fbrep) | (NOT (SIZEOF(
                QUERY ( fcs <* csh\connected_face_set.cfs_faces | (NOT (
                SIZEOF(QUERY ( bnds <* fcs.bounds | (
                'TECHNICAL_DATA_PACKAGING.FACE_OUTER_BOUND' IN TYPEOF(bnds))
))
                = 1)) )) = 0)) )) = 0)) )) = 0);
      wr5: (SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MANIFOLD_SOLID_BREP' IN
TYPEOF(it)) )
                 | ('TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL' IN
                TYPEOF(msb\manifold_solid_brep.outer)) )) = 0);
      wr6: (SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.BREP_WITH_VOIDS' IN TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids |
                csh\oriented_closed_shell.orientation )) = 0)) )) = 0);
      wr7: (SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT ('TECHNICAL_DATA_PACKAGING.' +
                     'FACETED_BREP_SHAPE_REPRESENTATION'
                IN TYPEOF(mi\mapped_item.mapping_source.
                mapped_representation))) )) = 0);
  END_ENTITY; -- faceted_brep_shape_representation

  ENTITY fill_area_style;
      name        : label;
      fill_styles : SET [1:?] OF fill_style_select;
    WHERE
      wr1: (SIZEOF(QUERY ( fill_style <* SELF.fill_styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'FILL_AREA_STYLE_COLOUR') IN
                TYPEOF(fill_style)) )) <= 1);
  END_ENTITY; -- fill_area_style

  ENTITY fill_area_style_colour;
      name        : label;
      fill_colour : colour;
  END_ENTITY; -- fill_area_style_colour

  ENTITY founded_item;
  END_ENTITY; -- founded_item
```



```
ENTITY functionally_defined_transformation;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- functionally_defined_transformation

ENTITY general_property;
    id          : identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- general_property

ENTITY general_property_association;
    name               : label;
    description        : OPTIONAL text;
    base_definition    : general_property;
    derived_definition : derived_property_select;
  WHERE
    wr1: (SIZEOF(USEDIN(derived_definition,'TECHNICAL_DATA_PACKAGING.' +
              'GENERAL_PROPERTY_ASSOCIATION.' + 'DERIVED_DEFINITION')) =
             1);
    wr2: (derived_definition.name = base_definition.name);
END_ENTITY; -- general_property_association

ENTITY general_property_relationship;
    name              : label;
    description       : OPTIONAL text;
    relating_property : general_property;
    related_property  : general_property;
END_ENTITY; -- general_property_relationship

ENTITY geometric_curve_set
  SUBTYPE OF (geometric_set);
  WHERE
    wr1: (SIZEOF(QUERY ( temp <* SELF\geometric_set.elements | (
              'TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(temp)) )) = 0);
END_ENTITY; -- geometric_curve_set

ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
    coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (point,direction,vector,placement,
      cartesian_transformation_operator,curve,surface,edge_curve,
      face_surface,poly_loop,vertex_point,solid_model,boolean_result,
      sphere,right_circular_cone,right_circular_cylinder,torus,block,
      right_angular_wedge,half_space_solid,shell_based_surface_model,
      face_based_surface_model,shell_based_wireframe_model,
      edge_based_wireframe_model,geometric_set))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    wr1: (SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (
              NOT (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_CONTEXT'
              IN TYPEOF(using_rep.context_of_items))) )) = 0);
END_ENTITY; -- geometric_representation_item
```

```
  ENTITY geometric_set
    SUPERTYPE OF (ONEOF (geometric_curve_set))
    SUBTYPE OF (geometric_representation_item);
      elements : SET [1:?] OF geometric_set_select;
  END_ENTITY; -- geometric_set

  ENTITY geometrically_bounded_2d_wireframe_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SELF.context_of_items\geometric_representation_context.
           coordinate_space_dimension = 2);
      wr2: (SIZEOF(QUERY ( item <* SELF.items | (NOT (SIZEOF(TYPEOF(item)
           * ['TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET',
           'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_2D',
           'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM']) = 1)) )) = 0);
      wr3: (SIZEOF(QUERY ( item <* SELF.items | (SIZEOF(TYPEOF(item) * [
           'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET',
           'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM']) = 1) )) >= 1);
      wr4: (SIZEOF(QUERY ( mi <* QUERY ( item <* SELF.items | (
           'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(item)) ) |
           (NOT (('TECHNICAL_DATA_PACKAGING.' +
           'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION') IN
           TYPEOF(mi\mapped_item.mapping_source
            .mapped_representation))) ))
           = 0);
      wr5: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
           'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN TYPEOF(
           item)) ) | (NOT (SIZEOF(QUERY ( elem <* gcs\geometric_set.
           elements | (NOT (SIZEOF(TYPEOF(elem) * [
           'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
           'TECHNICAL_DATA_PACKAGING.CIRCLE',
           'TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE',
           'TECHNICAL_DATA_PACKAGING.ELLIPSE',
           'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_2D',
           'TECHNICAL_DATA_PACKAGING.POINT',
           'TECHNICAL_DATA_PACKAGING.POLYLINE',
           'TECHNICAL_DATA_PACKAGING.TRIMMED_CURVE']) = 1)) )) = 0)) ))
           = 0);
      wr6: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
           'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN TYPEOF(
           item)) ) | (NOT (SIZEOF(QUERY ( crv <* QUERY ( elem <* gcs\
           geometric_set.elements | ('TECHNICAL_DATA_PACKAGING.CURVE'
           IN TYPEOF(elem)) ) | (NOT valid_basis_curve_in_2d_wireframe(
           crv)) )) = 0)) )) = 0);
      wr7: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items |
           ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET'
           IN TYPEOF(item)) ) | (NOT (SIZEOF(QUERY ( pnt <*
           QUERY ( elem <* gcs\geometric_set.elements | (
           'TECHNICAL_DATA_PACKAGING.POINT' IN TYPEOF(elem)) ) | (NOT (
           SIZEOF(TYPEOF(pnt) * [
           'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT',
           'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE']) = 1)) )) = 0))
))
           = 0);
      wr8: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
           'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN TYPEOF(
           item)) ) | (NOT (SIZEOF(QUERY ( pl <* QUERY ( elem <* gcs\
           geometric_set.elements | (
           'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(elem)) ) | (
```



```
              NOT (SIZEOF(pl\polyline.points) > 2)) )) = 0)) )) = 0);
  END_ENTITY; -- geometrically_bounded_2d_wireframe_representation

  ENTITY geometrically_bounded_surface_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
              = 1)) )) = 0);
      wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
              > 0);
      wr3: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
              NOT ((('TECHNICAL_DATA_PACKAGING.' +
              'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION') IN
              TYPEOF(mi\mapped_item.mapping_source.mapped_representation))
              AND (SIZEOF(QUERY ( mr_it <* mi\mapped_item.mapping_source.
              mapped_representation.items | (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET' IN TYPEOF(mr_it))
))
              > 0))) )) = 0);
      wr4: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET' IN TYPEOF(it)) ) |
              (NOT (SIZEOF(QUERY ( pnt <* QUERY ( gsel <* gs\geometric_set
              .elements | ('TECHNICAL_DATA_PACKAGING.POINT' IN
TYPEOF(gsel)) )
               | (NOT gbsf_check_point(pnt)) )) = 0)) )) = 0);
      wr5: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET' IN TYPEOF(it)) ) |
              (NOT (SIZEOF(QUERY ( cv <* QUERY ( gsel <* gs\geometric_set.
              elements | ('TECHNICAL_DATA_PACKAGING.CURVE' IN
TYPEOF(gsel)) )
               | (NOT gbsf_check_curve(cv)) )) = 0)) )) = 0);
      wr6: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET' IN TYPEOF(it)) ) |
              (NOT (SIZEOF(QUERY ( sf <* QUERY ( gsel <* gs\geometric_set.
              elements | ('TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(
              gsel)) ) | (NOT gbsf_check_surface(sf)) )) = 0)) )) = 0);
      wr7: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_SET' IN TYPEOF(it)) ) |
              (SIZEOF(QUERY ( gsel <* gs\geometric_set.elements | (
              'TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(gsel)) )) > 0)
))
              > 0);
  END_ENTITY; -- geometrically_bounded_surface_shape_representation

  ENTITY geometrically_bounded_wireframe_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF(TYPEOF(it) * [
              'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D',
              'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM']) = 1)) )) = 0);
      wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF(TYPEOF(it) * [
```

```
                'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM']) = 1) )) >= 1);
      wr3: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( crv <* QUERY ( elem <* gcs\
                geometric_set.elements | ('TECHNICAL_DATA_PACKAGING.CURVE'
                IN TYPEOF(elem)) ) | (NOT
                valid_geometrically_bounded_wf_curve(crv)) )) = 0)) )) = 0);
      wr4: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( pnts <* QUERY ( elem <* gcs\
                geometric_set.elements | ('TECHNICAL_DATA_PACKAGING.POINT'
                IN TYPEOF(elem)) ) | (NOT
                valid_geometrically_bounded_wf_point(pnts)) )) = 0)) )) =
0);
      wr5: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( cnc <* QUERY ( elem <* gcs\
                geometric_set.elements | ('TECHNICAL_DATA_PACKAGING.CONIC'
                IN TYPEOF(elem)) ) | (NOT (
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D' IN TYPEOF(cnc\
                conic.position))) )) = 0)) )) = 0);
      wr6: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.GEOMETRIC_CURVE_SET' IN
TYPEOF(it)) )
                 | (NOT (SIZEOF(QUERY ( pline <* QUERY ( elem <* gcs\
                geometric_set.elements | (
                'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(elem)) ) | (
                NOT (SIZEOF(pline\polyline.points) > 2)) )) = 0)) )) = 0);
      wr7: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT (('TECHNICAL_DATA_PACKAGING.' +
                'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION') IN
                TYPEOF(mi\mapped_item.mapping_source
                .mapped_representation))) ))
                = 0);
  END_ENTITY; -- geometrically_bounded_wireframe_shape_representation

  ENTITY global_uncertainty_assigned_context
    SUBTYPE OF (representation_context);
      uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
  END_ENTITY; -- global_uncertainty_assigned_context

  ENTITY global_unit_assigned_context
    SUBTYPE OF (representation_context);
      units : SET [1:?] OF unit;
  END_ENTITY; -- global_unit_assigned_context

  ENTITY group;
      name        : label;
      description : OPTIONAL text;
    DERIVE
      id : identifier := get_id_value(SELF);
```



```
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
END_ENTITY; -- group

ENTITY group_relationship;
    name           : label;
    description    : OPTIONAL text;
    relating_group : group;
    related_group  : group;
END_ENTITY; -- group_relationship

ENTITY half_space_solid
  SUBTYPE OF (geometric_representation_item);
    base_surface   : surface;
    agreement_flag : BOOLEAN;
END_ENTITY; -- half_space_solid

ENTITY hyperbola
  SUBTYPE OF (conic);
    semi_axis      : positive_length_measure;
    semi_imag_axis : positive_length_measure;
END_ENTITY; -- hyperbola

ENTITY id_attribute;
    attribute_value : identifier;
    identified_item : id_attribute_select;
END_ENTITY; -- id_attribute

ENTITY identification_assignment
  ABSTRACT SUPERTYPE;
    assigned_id : identifier;
    role        : identification_role;
END_ENTITY; -- identification_assignment

ENTITY identification_assignment_relationship;
    name                               : label;
    description                        : OPTIONAL text;
    relating_identification_assignment : identification_assignment;
    related_identification_assignment  : identification_assignment;
END_ENTITY; -- identification_assignment_relationship

ENTITY identification_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- identification_role

ENTITY intersection_curve
  SUBTYPE OF (surface_curve);
  WHERE
    wr1: (SIZEOF(SELF\surface_curve.associated_geometry) = 2);
    wr2: (associated_surface(SELF\surface_curve.associated_geometry[1])
              <> associated_surface(SELF\surface_curve.associated_geometry
              [2]));
END_ENTITY; -- intersection_curve

ENTITY invisibility;
    invisible_items : SET [1:?] OF invisible_item;
END_ENTITY; -- invisibility
```

```
ENTITY item_defined_transformation;
    name             : label;
    description      : OPTIONAL text;
    transform_item_1 : representation_item;
    transform_item_2 : representation_item;
END_ENTITY; -- item_defined_transformation

ENTITY item_identified_representation_usage;
    name                : label;
    description         : OPTIONAL text;
    definition          : represented_definition;
    used_representation : representation;
    identified_item     : representation_item;
  WHERE
    wr1: (SELF.used_representation IN using_representations(SELF.
            identified_item));
END_ENTITY; -- item_identified_representation_usage

ENTITY language_assignment
  SUBTYPE OF (classification_assignment);
    items : SET [1:?] OF language_item;
END_ENTITY; -- language_assignment

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.LENGTH_UNIT' IN TYPEOF(SELF\
            measure_with_unit.unit_component));
END_ENTITY; -- length_measure_with_unit

ENTITY length_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 1) AND (SELF\
            named_unit.dimensions.mass_exponent = 0) AND (SELF\
            named_unit.dimensions.time_exponent = 0) AND (SELF\
            named_unit.dimensions.electric_current_exponent = 0) AND (
            SELF\named_unit.dimensions.
            thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
            .dimensions.amount_of_substance_exponent = 0) AND (SELF\
            named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- length_unit

ENTITY library_context
  SUBTYPE OF (application_context_element);
    library_reference : label;
END_ENTITY; -- library_context

ENTITY light_source
  SUPERTYPE OF (ONEOF (light_source_ambient,light_source_directional,
      light_source_positional,light_source_spot))
  SUBTYPE OF (geometric_representation_item);
    light_colour : colour;
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'STYLED_ITEM.ITEM')) = 0);
END_ENTITY; -- light_source
```



```
ENTITY light_source_ambient
  SUBTYPE OF (light_source);
END_ENTITY; -- light_source_ambient

ENTITY light_source_directional
  SUBTYPE OF (light_source);
    orientation : direction;
END_ENTITY; -- light_source_directional

ENTITY light_source_positional
  SUBTYPE OF (light_source);
    position             : cartesian_point;
    constant_attenuation : REAL;
    distance_attenuation : REAL;
END_ENTITY; -- light_source_positional

ENTITY light_source_spot
  SUBTYPE OF (light_source);
    position              : cartesian_point;
    orientation           : direction;
    concentration_exponent : REAL;
    constant_attenuation   : REAL;
    distance_attenuation   : REAL;
    spread_angle          : positive_plane_angle_measure;
END_ENTITY; -- light_source_spot

ENTITY line
  SUBTYPE OF (curve);
    pnt : cartesian_point;
    dir : vector;
  WHERE
    wr1: (dir.dim = pnt.dim);
END_ENTITY; -- line

ENTITY local_time;
    hour_component   : hour_in_day;
    minute_component : OPTIONAL minute_in_hour;
    second_component : OPTIONAL second_in_minute;
    zone             : coordinated_universal_time_offset;
  WHERE
    wr1: valid_time(SELF);
END_ENTITY; -- local_time

ENTITY loop
  SUPERTYPE OF (ONEOF (vertex_loop,edge_loop,poly_loop))
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
    effectivity_lot_id   : identifier;
    effectivity_lot_size : measure_with_unit;
END_ENTITY; -- lot_effectivity
```

```
  ENTITY luminous_intensity_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.LUMINOUS_INTENSITY_UNIT' IN TYPEOF(
                SELF\measure_with_unit.unit_component));
  END_ENTITY; -- luminous_intensity_measure_with_unit

  ENTITY luminous_intensity_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
                named_unit.dimensions.mass_exponent = 0) AND (SELF\
                named_unit.dimensions.time_exponent = 0) AND (SELF\
                named_unit.dimensions.electric_current_exponent = 0) AND (
                SELF\named_unit.dimensions.
                thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
                .dimensions.amount_of_substance_exponent = 0) AND (SELF\
                named_unit.dimensions.luminous_intensity_exponent = 1));
  END_ENTITY; -- luminous_intensity_unit

  ENTITY make_from_usage_option
    SUBTYPE OF (product_definition_usage);
      ranking           : INTEGER;
      ranking_rationale : text;
      quantity          : measure_with_unit;
    WHERE
      wr1: ((NOT ('NUMBER' IN TYPEOF(quantity.value_component))) OR (
                quantity.value_component > 0));
  END_ENTITY; -- make_from_usage_option

  ENTITY make_from_usage_option_with_reference_designator
    SUBTYPE OF (make_from_usage_option, assembly_component_usage);
  END_ENTITY; -- make_from_usage_option_with_reference_designator

  ENTITY manifold_solid_brep
    SUBTYPE OF (solid_model);
      outer : closed_shell;
  END_ENTITY; -- manifold_solid_brep

  ENTITY manifold_surface_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] *
TYPEOF(it))
                = 1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
                > 0);
      wr3 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT(('TECHNICAL_DATA_PACKAGING.' +
                     'MANIFOLD_SURFACE_SHAPE_REPRESENTATION'
                IN TYPEOF(mi\mapped_item.mapping_source.
                mapped_representation)) AND (SIZEOF(QUERY ( mr_it <* mi\
```



```
                mapped_item.mapping_source.mapped_representation.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(mr_it)) )) > 0))) )) = 0);
      wr4 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( sh <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.OPEN_SHELL',
                'TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL',
                'TECHNICAL_DATA_PACKAGING.CLOSED_SHELL'] * TYPEOF(sh)) =
1)) ))
                = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT (
                'TECHNICAL_DATA_PACKAGING.FACE_SURFACE' IN TYPEOF(fa))) ))
                = 0)) )) = 0)) )) = 0);
      wr6 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                msf_surface_check(fa\face_surface.face_geometry))) )) = 0))
))
                = 0)) )) = 0);
      wr7 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( bnds <* fa.bounds | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP',
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP'] *
TYPEOF(bnds.bound))
                = 1)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
      wr8 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
                NOT ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN TYPEOF(oe.
                edge_element))) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) =
0);
      wr9 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
```

```
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe_cv <* QUERY ( oe <* elp_fbnds\
               path.edge_list | ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN
               TYPEOF(oe.edge_element)) ) | (NOT (SIZEOF([
               'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
               'TECHNICAL_DATA_PACKAGING.CONIC',
               'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA',
               'TECHNICAL_DATA_PACKAGING.LINE',
               'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D',
               'TECHNICAL_DATA_PACKAGING.PCURVE',
               'TECHNICAL_DATA_PACKAGING.POLYLINE',
               'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE'] * TYPEOF(oe_cv.
               edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
                0))) )) = 0)) )) = 0)) )) = 0);
     wr10: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
               shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
               QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
               'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
               (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
               'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
               NOT msf_curve_check(oe.edge_element\edge_curve.
               edge_geometry)) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) =
0);
     wr11: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
               shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
               QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
               'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
               (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
               'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
               NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe.
               edge_element.edge_start)) AND (
               'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe.
               edge_element.edge_end)))) )) = 0)) )) = 0))) )) = 0)) )) =
               0)) )) = 0);
     wr12: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
               'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
               TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
               shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
               QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
               'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
               (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
               'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
               NOT ((SIZEOF(['TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT',
               'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
               'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
               'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(oe.
               edge_element.edge_start\vertex_point.vertex_geometry)) = 1)
               AND (SIZEOF(['TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT',
```



```
                'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(oe.
                edge_element.edge_end\vertex_point.vertex_geometry)) = 1)))
))
                = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
      wr13: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN
TYPEOF(bnds.bound)) )
                 | (NOT ('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(
                vlp_fbnds\vertex_loop.loop_vertex))) )) = 0))) )) = 0)) ))
                = 0)) )) = 0);
      wr14: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
                shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
                QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN
TYPEOF(bnds.bound)) )
                 | (NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING' +
                '.CARTESIAN_POINT',
                'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(
                vlp_fbnds\vertex_loop.loop_vertex\vertex_point.
                vertex_geometry)) = 1)) )) = 0))) )) = 0)) )) = 0)) )) =
0);
  END_ENTITY; -- manifold_surface_shape_representation

  ENTITY mapped_item
    SUBTYPE OF (representation_item);
      mapping_source : representation_map;
      mapping_target : representation_item;
    WHERE
      wr1: acyclic_mapped_representation(using_representations(SELF),
          [SELF]);
  END_ENTITY; -- mapped_item

  ENTITY mass_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.MASS_UNIT' IN TYPEOF(SELF\
                measure_with_unit.unit_component));
  END_ENTITY; -- mass_measure_with_unit

  ENTITY mass_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
                named_unit.dimensions.mass_exponent = 1) AND (SELF\
                named_unit.dimensions.time_exponent = 0) AND (SELF\
```



```
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
  END_ENTITY; -- mass_unit

  ENTITY material_designation;
      name        : label;
      definitions : SET [1:?] OF characterized_definition;
  END_ENTITY; -- material_designation

  ENTITY measure_qualification;
      name              : label;
      description       : text;
      qualified_measure : measure_with_unit;
      qualifiers        : SET [1:?] OF value_qualifier;
    WHERE
      wr1: (SIZEOF(QUERY ( temp <* qualifiers | (
              'TECHNICAL_DATA_PACKAGING.PRECISION_QUALIFIER' IN TYPEOF(
              temp)) )) < 2);
  END_ENTITY; -- measure_qualification

  ENTITY measure_representation_item
    SUBTYPE OF (representation_item, measure_with_unit);
  END_ENTITY; -- measure_representation_item

  ENTITY measure_with_unit
    SUPERTYPE OF (ONEOF (length_measure_with_unit,mass_measure_with_unit,
        time_measure_with_unit,electric_current_measure_with_unit,
        thermodynamic_temperature_measure_with_unit,
        amount_of_substance_measure_with_unit,
        luminous_intensity_measure_with_unit,plane_angle_measure_with_unit,
        solid_angle_measure_with_unit,area_measure_with_unit,
        volume_measure_with_unit,ratio_measure_with_unit));
      value_component : measure_value;
      unit_component  : unit;
    WHERE
      wr1: valid_units(SELF);
  END_ENTITY; -- measure_with_unit

  ENTITY mechanical_design_geometric_presentation_area
    SUBTYPE OF (presentation_area);
    WHERE
      wr1: (SIZEOF(QUERY ( it1 <* SELF.items | (NOT ((
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(it1))
              OR (('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it1))
              AND ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(
              it1\mapped_item.mapping_source.mapped_representation))))) ))
              = 0);
      wr2: (SIZEOF(QUERY ( pv <* QUERY ( mi1 <* QUERY ( it1 <* SELF.items
               | ('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it1)) )
               | ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(
              mi1\mapped_item.mapping_source.mapped_representation)) ) | (
              NOT (SIZEOF(QUERY ( it2 <* pv\mapped_item.mapping_source.
              mapped_representation\representation.items | ((NOT ((
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(it2))
              OR (('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it2))
              AND (NOT (('TECHNICAL_DATA_PACKAGING.' +
```



```
                'CAMERA_IMAGE_3D_WITH_SCALE') IN TYPEOF(it2))) AND (NOT (
                'TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(it2\
                mapped_item.mapping_source.mapped_representation)))))) OR ((
                ('TECHNICAL_DATA_PACKAGING.' + 'CAMERA_IMAGE_3D_WITH_SCALE')
                IN TYPEOF(it2)) AND (NOT (('TECHNICAL_DATA_PACKAGING.' +
                'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION')
                IN TYPEOF(it2\mapped_item.mapping_source.
                mapped_representation))))) )) = 0)) )) = 0);
      wr3: ((SIZEOF(QUERY ( ps <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') | (
                (‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’ IN TYPEOF(ps))
                AND (ps.size\planar_extent.size_in_x <= 0) OR (ps.size\
                planar_extent.size_in_y <= 0)) )) = 0) AND (SIZEOF(
                QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'AREA_IN_SET.AREA') |
                (‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’
                IN TYPEOF(ais)) AND (SIZEOF(
                QUERY ( ps <* USEDIN(ais,'TECHNICAL_DATA_PACKAGING.' +
                'PRESENTATION_SIZE.UNIT') | ((ps.size\planar_extent.
                size_in_x <= 0) OR (ps.size\planar_extent.size_in_y <= 0))
))
                > 0) )) = 0));
      wr4: (((SIZEOF(QUERY ( ps <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') |
                (‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’ IN TYPEOF(ps))
                AND (('TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_2D') IN
                TYPEOF(ps.size.placement)) )) = 1) AND (SIZEOF(
                QUERY ( ps <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') |
                (‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’ IN TYPEOF(ps))
                AND (('TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_3D') IN
                TYPEOF(ps.size.placement)) )) = 0)) OR ((SIZEOF(
                QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'AREA_IN_SET.AREA') |(
                ‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’ IN TYPEOF
                (ais)) AND (SIZEOF(
                QUERY ( ps <* USEDIN(ais,'TECHNICAL_DATA_PACKAGING.' +
                'PRESENTATION_SIZE.UNIT') | (('TECHNICAL_DATA_PACKAGING.' +
                'AXIS2_PLACEMENT_2D') IN TYPEOF(ps.size.placement)) )) = 1)
))
                = 1) AND (SIZEOF(QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'AREA_IN_SET.AREA') | (
                ‘TECHNICAL_DATA_PACKAGING.PRESENTATION_AREA’ IN TYPEOF(ais))
                AND (SIZEOF(QUERY ( ps <* USEDIN(ais,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') | ((
                'TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_3D') IN
                TYPEOF(ps.size.placement)) )) = 0) )) = 1)));
  END_ENTITY; -- mechanical_design_geometric_presentation_area

  ENTITY mechanical_design_geometric_presentation_representation
    SUBTYPE OF (representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT',
                'TECHNICAL_DATA_PACKAGING.CAMERA_MODEL_D3'] * TYPEOF(it)) =
                 1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
```

```
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
                'SHAPE_REPRESENTATION','TECHNICAL_DATA_PACKAGING.' +
                'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION']
                * TYPEOF(mi\mapped_item.mapping_source.
                mapped_representation)) = 1)) )) = 0);
      wr3 : (SIZEOF(QUERY ( smi <* QUERY ( si <* QUERY ( it <* SELF.items
                 | ('TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) )
                 | ('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(si\
                styled_item.item)) ) | (NOT (('TECHNICAL_DATA_PACKAGING.' +
                 'SHAPE_REPRESENTATION') IN TYPEOF(smi\styled_item.item\
                mapped_item.mapping_source.mapped_representation))) )) =
0);
      wr4 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( pss <* psa.styles | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.POINT_STYLE',
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE',
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE'] * TYPEOF(
                pss)) = 1)) )) = 0)) )) = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psbc <* QUERY ( psa <* si\styled_item.
                styles | (('TECHNICAL_DATA_PACKAGING.' +
                'PRESENTATION_STYLE_BY_CONTEXT') IN TYPEOF(psa)) ) | (NOT (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
                 'REPRESENTATION_ITEM',
                'TECHNICAL_DATA_PACKAGING.REPRESENTATION'] * TYPEOF(psbc\
                presentation_style_by_context.style_context)) = 1)) )) =
0)) ))
                = 0);
      wr6 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ps <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.POINT_STYLE' IN TYPEOF(pss)) ) |
                (NOT ((('TECHNICAL_DATA_PACKAGING.' +
                'POSITIVE_LENGTH_MEASURE') IN TYPEOF(ps\point_style.
                marker_size)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ps\point_style.
                marker_colour)) = 1))) )) = 0)) )) = 0)) )) = 0);
      wr7 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( cs <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(pss)) ) |
                (NOT ((SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(cs\curve_style.
                curve_colour)) = 1) AND (('TECHNICAL_DATA_PACKAGING.' +
                'POSITIVE_LENGTH_MEASURE') IN TYPEOF(cs\curve_style.
                curve_width)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(cs\
                curve_style.curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0);
```



```
      wr8 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (('TECHNICAL_DATA_PACKAGING.' +
                'SURFACE_SIDE_STYLE') IN TYPEOF(ssu\surface_style_usage.
                style))) )) = 0)) )) = 0)) )) = 0);
      wr9 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | (NOT
                (SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
                'SURFACE_STYLE_PARAMETER_LINE','TECHNICAL_DATA_PACKAGING.'
                + 'SURFACE_STYLE_CONTROL_GRID','TECHNICAL_DATA_PACKAGING.'
                + 'SURFACE_STYLE_SILHOUETTE','TECHNICAL_DATA_PACKAGING.' +
                'SURFACE_STYLE_SEGMENTATION_CURVE',
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_FILL_AREA',
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_BOUNDARY'] *
                TYPEOF(sses)) = 1)) )) = 0)) )) = 0)) )) = 0)) )) = 0);
      wr10: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( sspl <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' +
                'SURFACE_STYLE_PARAMETER_LINE') IN TYPEOF(sses)) ) | (NOT (
                ('TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sspl\
                surface_style_parameter_line.style_of_parameter_lines)) AND
                 (SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sspl\
                surface_style_parameter_line.style_of_parameter_lines\
                curve_style.curve_colour)) = 1) AND ((
                'TECHNICAL_DATA_PACKAGING.' + 'POSITIVE_LENGTH_MEASURE') IN
                 TYPEOF(sspl\surface_style_parameter_line.
                style_of_parameter_lines\curve_style.curve_width)) AND (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sspl\
                surface_style_parameter_line.style_of_parameter_lines\
                curve_style.curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0))
))
                = 0);
      wr11: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( sscg <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_CONTROL_GRID')
                IN TYPEOF(sses)) ) | (NOT ((
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sscg\
```

```
                surface_style_control_grid.style_of_control_grid)) AND (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sscg\
                surface_style_control_grid.style_of_control_grid\
                curve_style.curve_colour)) = 1) AND ((
                'TECHNICAL_DATA_PACKAGING.' + 'POSITIVE_LENGTH_MEASURE') IN
                 TYPEOF(sscg\surface_style_control_grid.
                style_of_control_grid\curve_style.curve_width)) AND (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sscg\
                surface_style_control_grid.style_of_control_grid\
                curve_style.curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0))
))
                = 0);
     wr12: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( sssh <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_SILHOUETTE')
                IN TYPEOF(sses)) ) | (NOT ((
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sssh\
                surface_style_silhouette.style_of_silhouette)) AND (SIZEOF(
                ['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssh\
                surface_style_silhouette.style_of_silhouette\curve_style.
                curve_colour)) = 1) AND (('TECHNICAL_DATA_PACKAGING.' +
                'POSITIVE_LENGTH_MEASURE') IN TYPEOF(sssh\
                surface_style_silhouette.style_of_silhouette\curve_style.
                curve_width)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sssh\
                surface_style_silhouette.style_of_silhouette\curve_style.
                curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
     wr13: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( sssc <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' +
                'SURFACE_STYLE_SEGMENTATION_CURVE') IN TYPEOF(sses)) ) | (
                NOT (('TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sssc
                \surface_style_segmentation_curve.
                style_of_segmentation_curve)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssc\
                surface_style_segmentation_curve.
                style_of_segmentation_curve\curve_style.curve_colour)) = 1)
                AND (('TECHNICAL_DATA_PACKAGING.' +
                'POSITIVE_LENGTH_MEASURE') IN TYPEOF(sssc\
```



```
                surface_style_segmentation_curve.
                style_of_segmentation_curve\curve_style.curve_width)) AND (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sssc\
                surface_style_segmentation_curve.
                style_of_segmentation_curve\curve_style.curve_font)) = 1)))
))
                = 0)) )) = 0)) )) = 0)) )) = 0);
      wr14: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( ssbd <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_BOUNDARY') IN
                TYPEOF(sses)) ) | (NOT ((
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\curve_style.
                curve_colour)) = 1) AND (('TECHNICAL_DATA_PACKAGING.' +
                'POSITIVE_LENGTH_MEASURE') IN TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\curve_style.
                curve_width)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\curve_style.
                curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
  END_ENTITY; -- mechanical_design_geometric_presentation_representation

  ENTITY mechanical_design_shaded_presentation_area
    SUBTYPE OF (presentation_area);
    WHERE
      wr1: (SIZEOF(QUERY ( it1 <* SELF.items | (NOT ((
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(it1))
                OR (('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it1))
                AND ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(
                it1\mapped_item.mapping_source.mapped_representation))))) ))
                = 0);
      wr2: (SIZEOF(QUERY ( pv <* QUERY ( mi1 <* QUERY ( it1 <* SELF.items
                 | ('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it1)) )
                 | ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(
                mi1\mapped_item.mapping_source.mapped_representation)) ) | (
                NOT (SIZEOF(QUERY ( it2 <* pv\mapped_item.mapping_source.
                mapped_representation\representation.items | (
                (‘TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW’ IN TYPEOF(pv))
                AND(NOT ((
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN TYPEOF(it2))
                OR (('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it2))
                AND (NOT (('TECHNICAL_DATA_PACKAGING.' +
                'CAMERA_IMAGE_3D_WITH_SCALE') IN TYPEOF(it2))) AND (NOT (
                'TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(it2\
                mapped_item.mapping_source.mapped_representation)))))) OR ((
                ('TECHNICAL_DATA_PACKAGING.' + 'CAMERA_IMAGE_3D_WITH_SCALE')
```

```
                IN TYPEOF(it2)) AND (NOT (('TECHNICAL_DATA_PACKAGING.' +
                'MECHANICAL_DESIGN_SHADED_PRESENTATION_REPRESENTATION') IN
                TYPEOF(it2\mapped_item.mapping_source
                 .mapped_representation))
                ))) ))  = 0)) )) = 0);
      wr3: ((SIZEOF(QUERY ( ps <* USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.'
                + 'PRESENTATION_SIZE.UNIT') | (NOT ((ps.size\planar_extent.
                size_in_x > 0) AND (ps.size\planar_extent.size_in_y > 0)))
))
                = 0) AND (SIZEOF(QUERY ( pset <* QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'AREA_IN_SET.AREA') | ((
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SET') IN TYPEOF(
                ais.in_set)) ) | (SIZEOF(QUERY ( psize <* USEDIN(pset,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') | (
                NOT ((psize.size\planar_extent.size_in_x > 0) AND (psize.
                size\planar_extent.size_in_y > 0))) )) = 0) )) = 0));
      wr4: ((SIZEOF(QUERY ( psize <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') | ((
                'TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_2D') IN
                TYPEOF(psize.size.placement)) )) = 1) AND (SIZEOF(
                QUERY ( pset <* QUERY ( ais <* USEDIN(SELF,
                'TECHNICAL_DATA_PACKAGING.' + 'AREA_IN_SET.AREA') | ((
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SET') IN TYPEOF(
                ais.in_set)) ) | (SIZEOF(QUERY ( psize <* USEDIN(pset,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT') | (
                NOT (('TECHNICAL_DATA_PACKAGING.' + 'AXIS2_PLACEMENT_2D') IN
                 TYPEOF(psize.size.placement))) )) = 0) )) = 0));
      wr5: (SIZEOF(QUERY ( pv <* QUERY ( mi1 <* QUERY ( it1 <* SELF.items
                 | ('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it1)) )
                 | ('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN TYPEOF(
                mi1\mapped_item.mapping_source.mapped_representation)) ) | (
                NOT (SIZEOF(QUERY ( ci <* pv\mapped_item.mapping_source.
                mapped_representation\representation.items |
                (('TECHNICAL_DATA_PACKAGING.PRESENTATION_VIEW' IN
                TYPEOF(pv)) AND((
                'TECHNICAL_DATA_PACKAGING.' + 'CAMERA_IMAGE_3D_WITH_SCALE')
                IN TYPEOF(ci)) AND (SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
                'CAMERA_MODEL_D3','TECHNICAL_DATA_PACKAGING.' +
                'CAMERA_MODEL_D3_WITH_HLHSR','TECHNICAL_DATA_PACKAGING.' +
                'CAMERA_MODEL_WITH_LIGHT_SOURCES'] * TYPEOF(ci\
                mapped_item.mapping_source.mapping_origin)) =
                 1)) )) = 0)) )) = 0);
  END_ENTITY; -- mechanical_design_shaded_presentation_area

  ENTITY mechanical_design_shaded_presentation_representation
    SUBTYPE OF (representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                 'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                 'TECHNICAL_DATA_PACKAGING.STYLED_ITEM',
                 'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT',
                 'TECHNICAL_DATA_PACKAGING.CAMERA_MODEL_D3'] * TYPEOF(it)) =
                  1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                 'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
                'SHAPE_REPRESENTATION','TECHNICAL_DATA_PACKAGING.' +
                'MECHANICAL_DESIGN_SHADED_PRESENTATION_REPRESENTATION'] *
                TYPEOF(mi\mapped_item.mapping_source
```



```
                .mapped_representation))
                = 1)) )) = 0);
      wr3 : (SIZEOF(QUERY ( smi <* QUERY ( si <* QUERY ( it <* SELF.items
                 | ('TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) )
                 | ('TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(si\
                styled_item.item)) ) | (NOT (('TECHNICAL_DATA_PACKAGING.' +
                 'SHAPE_REPRESENTATION') IN TYPEOF(smi\styled_item.item\
                mapped_item.mapping_source.mapped_representation))) )) =
0);
      wr4 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( pss <* psa.styles | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.POINT_STYLE',
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE',
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE'] * TYPEOF(
                pss)) = 1)) )) = 0)) )) = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psbc <* QUERY ( psa <* si\styled_item.
                styles | (('TECHNICAL_DATA_PACKAGING.' +
                'PRESENTATION_STYLE_BY_CONTEXT') IN TYPEOF(psa)) ) | (NOT (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.REPRESENTATION_ITEM',
                'TECHNICAL_DATA_PACKAGING.REPRESENTATION'] * TYPEOF(psbc\
                presentation_style_by_context.style_context)) = 1)) )) =
0)) ))
                = 0);
      wr6 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ps <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.POINT_STYLE' IN TYPEOF(pss)) ) |
                (NOT (('TECHNICAL_DATA_PACKAGING.MARKER_TYPE' IN TYPEOF(ps\
                point_style.marker)) AND (
                'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
                TYPEOF(ps\point_style.marker_size)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ps\point_style.
                marker_colour)) = 1))) )) = 0)) )) = 0)) )) = 0);
      wr7 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( cs <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(pss)) ) |
                (NOT ((SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(cs\curve_style.
                curve_colour)) = 1) AND (
                'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
                TYPEOF(cs\curve_style.curve_width)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(cs\
                curve_style.curve_font)) = 1))) )) = 0)) )) = 0)) )) = 0);
      wr8 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
```

```
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (
            'TECHNICAL_DATA_PACKAGING.SURFACE_SIDE_STYLE' IN TYPEOF(ssu
            \surface_style_usage.style))) )) = 0)) )) = 0)) )) = 0);
      wr9 : (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | (NOT
            (SIZEOF(['TECHNICAL_DATA_PACKAGING.' +
            'SURFACE_STYLE_PARAMETER_LINE','TECHNICAL_DATA_PACKAGING.'
            + 'SURFACE_STYLE_CONTROL_GRID','TECHNICAL_DATA_PACKAGING.'
            + 'SURFACE_STYLE_SILHOUETTE','TECHNICAL_DATA_PACKAGING.' +
            'SURFACE_STYLE_SEGMENTATION_CURVE',
            'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_BOUNDARY',
            'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_FILL_AREA',
            'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_RENDERING'] *
            TYPEOF(sses)) = 1)) )) = 0)) )) = 0)) )) = 0)) )) = 0);
      wr10: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( ssfa <* QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_FILL_AREA' IN
            TYPEOF(sses)) ) | (NOT (SIZEOF(QUERY ( fss <* ssfa\
            surface_style_fill_area.fill_area.fill_styles | (NOT (((
            'TECHNICAL_DATA_PACKAGING.' + 'FILL_AREA_STYLE_COLOUR') IN
            TYPEOF(fss)) AND (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(fss\
            fill_area_style_colour.fill_colour)) = 1))) )) = 0)) )) =
          0)) ))
            = 0)) )) = 0)) )) = 0);
      wr11: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( sspl <* QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | ((
            'TECHNICAL_DATA_PACKAGING.' +
            'SURFACE_STYLE_PARAMETER_LINE') IN TYPEOF(sses)) ) | (NOT (
            (('TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sspl\
            surface_style_parameter_line.style_of_parameter_lines)) AND
             (SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sspl\
            surface_style_parameter_line.style_of_parameter_lines\
            curve_style.curve_colour)) = 1) AND (
            'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
            TYPEOF(sspl\surface_style_parameter_line.
            style_of_parameter_lines\curve_style.curve_width)) AND (
            SIZEOF(['TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
            'TECHNICAL_DATA_PACKAGING.' +
```



```
            'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sspl\
            surface_style_parameter_line.style_of_parameter_lines\
            curve_style.curve_font)) = 1)) OR ((
            'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_RENDERING' IN TYPEOF(
            sspl\
            surface_style_parameter_line.style_of_parameter_lines))
            AND (SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sspl\
            surface_style_parameter_line.style_of_parameter_lines\
            curve_style_rendering.rendering_properties.
            rendered_colour))
            = 1)))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
wr12: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( sscg <* QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | ((
            'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_CONTROL_GRID')
            IN TYPEOF(sses)) ) | (NOT (((
            'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sscg\
            surface_style_control_grid.style_of_control_grid)) AND (
            SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sscg\
            surface_style_control_grid.style_of_control_grid\
            curve_style.curve_colour)) = 1) AND (
            'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
            TYPEOF(sscg\surface_style_control_grid.
            style_of_control_grid\curve_style.curve_width)) AND (
            SIZEOF(['TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sscg\
            surface_style_control_grid.style_of_control_grid\
            curve_style.curve_font)) = 1)) OR ((
            'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_RENDERING' IN TYPEOF(
            sscg\surface_style_control_grid.style_of_control_grid)) AND
             (SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sscg\
            surface_style_control_grid.style_of_control_grid\
            curve_style_rendering.rendering_properties.
            rendered_colour))
            = 1)))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
wr13: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( sssh <* QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | ((
            'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_SILHOUETTE')
            IN TYPEOF(sses)) ) | (NOT (((
            'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(sssh\
            surface_style_silhouette.style_of_silhouette)) AND (SIZEOF(
            ['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
```

```
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssh\
            surface_style_silhouette.style_of_silhouette\curve_style.
            curve_colour)) = 1) AND (
            'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
            TYPEOF(sssh\surface_style_silhouette.style_of_silhouette\
            curve_style.curve_width)) AND (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sssh\
            surface_style_silhouette.style_of_silhouette\curve_style.
            curve_font)) = 1)) OR ((
            'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_RENDERING' IN TYPEOF(
            sssh\surface_style_silhouette.style_of_silhouette)) AND (
            SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssh\
            surface_style_silhouette.style_of_silhouette\
            curve_style_rendering.rendering_properties.
            rendered_colour))
            = 1)))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
    wr14: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
            'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
            pss)) ) | (NOT (SIZEOF(QUERY ( sssc <* QUERY ( sses <* ssu\
            surface_style_usage.style\surface_side_style.styles | ((
            'TECHNICAL_DATA_PACKAGING.' +
            'SURFACE_STYLE_SEGMENTATION_CURVE') IN TYPEOF(sses)) ) | (
            NOT ((('TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(
            sssc\surface_style_segmentation_curve.
            style_of_segmentation_curve)) AND (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssc\
            surface_style_segmentation_curve.
            style_of_segmentation_curve\curve_style.curve_colour)) = 1)
            AND ('TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
            TYPEOF(sssc\surface_style_segmentation_curve.
            style_of_segmentation_curve\curve_style.curve_width)) AND (
            SIZEOF(['TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sssc\
            surface_style_segmentation_curve.
            style_of_segmentation_curve\curve_style.curve_font)) = 1))
            OR (('TECHNICAL_DATA_PACKAGING.CURVE_STYLE_RENDERING' IN
            TYPEOF(sssc\surface_style_segmentation_curve.
            style_of_segmentation_curve)) AND (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
            'TECHNICAL_DATA_PACKAGING.' +
            'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssc\
            surface_style_segmentation_curve.
            style_of_segmentation_curve\curve_style_rendering.
            rendering_properties.rendered_colour)) = 1)))) )) = 0)) ))
            = 0)) )) = 0)) )) = 0);
    wr15: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
            'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
            NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
            SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
```



```
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( ssbd <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_BOUNDARY') IN
                TYPEOF(sses)) ) | (NOT (((
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE' IN TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\curve_style.
                curve_colour)) = 1) AND (
                'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN
                TYPEOF(ssbd\surface_style_boundary.style_of_boundary\
                curve_style.curve_width)) AND (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.' + 'CURVE_STYLE_FONT',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\curve_style.
                curve_font)) = 1)) OR ((
                'TECHNICAL_DATA_PACKAGING.CURVE_STYLE_RENDERING' IN TYPEOF(
                ssbd\surface_style_boundary.style_of_boundary)) AND (
                SIZEOF(['TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ssbd\
                surface_style_boundary.style_of_boundary\
                curve_style_rendering.rendering_properties
                .rendered_colour))
                = 1)))) )) = 0)) )) = 0)) )) = 0)) )) = 0);
    wr16: (SIZEOF(QUERY ( si <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.STYLED_ITEM' IN TYPEOF(it)) ) | (
                NOT (SIZEOF(QUERY ( psa <* si\styled_item.styles | (NOT (
                SIZEOF(QUERY ( ssu <* QUERY ( pss <* psa.styles | (
                'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
                pss)) ) | (NOT (SIZEOF(QUERY ( ssre <* QUERY ( sses <* ssu\
                surface_style_usage.style\surface_side_style.styles | ((
                'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_RENDERING') IN
                 TYPEOF(sses)) ) | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.COLOUR_RGB',
                'TECHNICAL_DATA_PACKAGING.' +
                'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ssre\
                surface_style_rendering.surface_colour)) = 1)) )) = 0)) ))
                = 0)) )) = 0)) )) = 0);
END_ENTITY; -- mechanical_design_shaded_presentation_representation

ENTITY name_attribute;
    attribute_value : label;
    named_item      : name_attribute_select;
END_ENTITY; -- name_attribute

ENTITY named_unit
  SUPERTYPE OF (ONEOF (si_unit,conversion_based_unit,
      context_dependent_unit) ANDOR ONEOF (length_unit,mass_unit,
      time_unit,electric_current_unit,thermodynamic_temperature_unit,
      amount_of_substance_unit,luminous_intensity_unit,plane_angle_unit,
      solid_angle_unit,area_unit,volume_unit,ratio_unit));
    dimensions : dimensional_exponents;
END_ENTITY; -- named_unit
```

```
  ENTITY next_assembly_usage_occurrence
    SUBTYPE OF (assembly_component_usage);
  END_ENTITY; -- next_assembly_usage_occurrence

  ENTITY non_manifold_surface_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] *
TYPEOF(it))
                = 1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
                > 0);
      wr3 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT ((('TECHNICAL_DATA_PACKAGING.' +
                'NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION') IN TYPEOF(mi\
                mapped_item.mapping_source.mapped_representation)) AND (
                SIZEOF(QUERY ( mr_it <* mi\mapped_item.mapping_source.
                mapped_representation.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(mr_it)) )) > 0))) )) = 0);
      wr4 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.FACE_SURFACE',
                'TECHNICAL_DATA_PACKAGING.ORIENTED_FACE'] * TYPEOF(fa)) =
1)) ))
                = 0)) )) = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( f_sf <* QUERY ( fa <* cfs.cfs_faces | (
                'TECHNICAL_DATA_PACKAGING.FACE_SURFACE' IN TYPEOF(fa)) ) |
                (NOT (('TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(
                f_sf)) OR nmsf_surface_check(f_sf\face_surface.
                face_geometry))) )) = 0)) )) = 0)) )) = 0);
      wr6 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( o_fa <* QUERY ( fa <* cfs.cfs_faces | (
                'TECHNICAL_DATA_PACKAGING.ORIENTED_FACE' IN TYPEOF(fa)) )
                 | (NOT (('TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN
                TYPEOF(o_fa\oriented_face.face_element)) OR
                nmsf_surface_check(o_fa\oriented_face.face_element\
                face_surface.face_geometry))) )) = 0)) )) = 0)) )) = 0);
      wr7 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
```



```
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( bnds <* fa.bounds | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP',
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP'] *
                 TYPEOF(bnds.bound))
                = 1)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
      wr8 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
                NOT ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN TYPEOF(oe.
                edge_element))) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) =
0);
      wr9 : (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe_cv <* QUERY ( oe <* elp_fbnds\
                path.edge_list | ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN
                TYPEOF(oe.edge_element)) ) | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
                'TECHNICAL_DATA_PACKAGING.CONIC',
                'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA',
                'TECHNICAL_DATA_PACKAGING.LINE',
                'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D',
                'TECHNICAL_DATA_PACKAGING.PCURVE',
                'TECHNICAL_DATA_PACKAGING.POLYLINE',
                'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE'] * TYPEOF(oe_cv.
                edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
                 0))) )) = 0)) )) = 0)) )) = 0);
      wr10: (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
                NOT nmsf_curve_check(oe.edge_element\edge_curve.
                edge_geometry)) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) =
0);
      wr11: (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
```



```
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
                NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe.
                edge_element.edge_start)) AND (
                'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(oe.
                edge_element.edge_end)))) )) = 0)) )) = 0))) )) = 0)) )) =
                0)) )) = 0);
      wr12: (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.EDGE_LOOP' IN TYPEOF(bnds.bound))
)
                 | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
                NOT ((SIZEOF(['TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT',
                'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(oe.
                edge_element.edge_start\vertex_point.vertex_geometry)) = 1)
                AND (SIZEOF(['TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT',
                'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(oe.
                edge_element.edge_end\vertex_point.vertex_geometry)) = 1)))
                ))
                = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
      wr13: (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN
TYPEOF(bnds.bound)) )
                 | (NOT ('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(
                vlp_fbnds\vertex_loop.loop_vertex))) )) = 0))) )) = 0)) ))
                = 0)) )) = 0);
      wr14: (SIZEOF(QUERY ( fbsm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.FACE_BASED_SURFACE_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* fbsm\
                face_based_surface_model.fbsm_faces | (NOT (SIZEOF(
                QUERY ( fa <* cfs.cfs_faces | (NOT ((
                'TECHNICAL_DATA_PACKAGING.ADVANCED_FACE' IN TYPEOF(fa)) OR
                (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_LOOP' IN
TYPEOF(bnds.bound)) )
                 | (NOT (SIZEOF(['TECHNICAL_DATA_PACKAGING' +
                 '.CARTESIAN_POINT',
                'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE',
                'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE'] * TYPEOF(
```



```
            vlp_fbnds\vertex_loop.loop_vertex\vertex_point.
            vertex_geometry)) = 1)) )) = 0))) )) = 0)) )) = 0)) )) =
          0);
END_ENTITY; -- non_manifold_surface_shape_representation

ENTITY object_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- object_role

ENTITY offset_curve_2d
  SUBTYPE OF (curve);
    basis_curve    : curve;
    distance       : length_measure;
    self_intersect : LOGICAL;
  WHERE
    wr1: (basis_curve.dim = 2);
END_ENTITY; -- offset_curve_2d

ENTITY offset_curve_3d
  SUBTYPE OF (curve);
    basis_curve    : curve;
    distance       : length_measure;
    self_intersect : LOGICAL;
    ref_direction  : direction;
  WHERE
    wr1: ((basis_curve.dim = 3) AND (ref_direction.dim = 3));
END_ENTITY; -- offset_curve_3d

ENTITY offset_surface
  SUBTYPE OF (surface);
    basis_surface  : surface;
    distance       : length_measure;
    self_intersect : LOGICAL;
END_ENTITY; -- offset_surface

ENTITY open_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell

ENTITY organization;
    id          : OPTIONAL identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- organization

ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
    assigned_organization : organization;
    role                  : organization_role;
END_ENTITY; -- organization_assignment

ENTITY organization_relationship;
    name                  : label;
    description           : OPTIONAL text;
    relating_organization : organization;
    related_organization  : organization;
END_ENTITY; -- organization_relationship
```

```
ENTITY organization_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- organization_role

ENTITY organizational_address
  SUBTYPE OF (address);
    organizations : SET [1:?] OF organization;
    description   : OPTIONAL text;
END_ENTITY; -- organizational_address

ENTITY organizational_project;
    name                     : label;
    description              : OPTIONAL text;
    responsible_organizations : SET [1:?] OF organization;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
END_ENTITY; -- organizational_project

ENTITY organizational_project_assignment
  ABSTRACT SUPERTYPE;
    assigned_organizational_project : organizational_project;
    role                            : organizational_project_role;
END_ENTITY; -- organizational_project_assignment

ENTITY organizational_project_relationship;
    name                            : label;
    description                     : OPTIONAL text;
    relating_organizational_project : organizational_project;
    related_organizational_project  : organizational_project;
END_ENTITY; -- organizational_project_relationship

ENTITY organizational_project_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- organizational_project_role

ENTITY oriented_closed_shell
  SUBTYPE OF (closed_shell);
    closed_shell_element : closed_shell;
    orientation          : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                          conditional_reverse(SELF.
                                          orientation,SELF.
                                          closed_shell_element
                                          .cfs_faces);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL' IN
              TYPEOF(SELF.closed_shell_element)));
END_ENTITY; -- oriented_closed_shell
```



```
ENTITY oriented_edge
  SUBTYPE OF (edge);
    edge_element : edge;
    orientation  : BOOLEAN;
  DERIVE
    SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_start,SELF.
                              edge_element.edge_end);
    SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_end,SELF.
                              edge_element.edge_start);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_EDGE' IN TYPEOF(SELF.
              edge_element)));
END_ENTITY; -- oriented_edge

ENTITY oriented_face
  SUBTYPE OF (face);
    face_element : face;
    orientation  : BOOLEAN;
  DERIVE
    SELF\face.bounds : SET [1:?] OF face_bound := conditional_reverse(
                          SELF.orientation,SELF.face_element.bounds);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_FACE' IN TYPEOF(SELF.
              face_element)));
END_ENTITY; -- oriented_face

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);
    open_shell_element : open_shell;
    orientation        : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                          conditional_reverse(SELF.
                                          orientation,SELF.
                                          open_shell_element.cfs_faces);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_OPEN_SHELL' IN TYPEOF(
              SELF.open_shell_element)));
END_ENTITY; -- oriented_open_shell

ENTITY oriented_path
  SUBTYPE OF (path);
    path_element : path;
    orientation  : BOOLEAN;
  DERIVE
    SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
                             conditional_reverse(SELF.orientation,SELF.
                             path_element.edge_list);
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.ORIENTED_PATH' IN TYPEOF(SELF.
              path_element)));
END_ENTITY; -- oriented_path

ENTITY oriented_surface
  SUBTYPE OF (surface);
  orientation   : BOOLEAN;
END_ENTITY; -- oriented_surface
```

```
ENTITY other_list_table_representation
  SUBTYPE OF (representation);
END_ENTITY; -- other_list_table_representation

ENTITY outer_boundary_curve
  SUBTYPE OF (boundary_curve);
END_ENTITY; -- outer_boundary_curve

ENTITY over_riding_styled_item
  SUBTYPE OF (styled_item);
    over_ridden_style : styled_item;
END_ENTITY; -- over_riding_styled_item

ENTITY parabola
  SUBTYPE OF (conic);
    focal_dist : length_measure;
  WHERE
    wr1: (focal_dist <> 0);
END_ENTITY; -- parabola

ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

ENTITY path
  SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
  SUBTYPE OF (topological_representation_item);
    edge_list : LIST [1:?] OF UNIQUE oriented_edge;
  WHERE
    wr1: path_head_to_tail(SELF);
END_ENTITY; -- path

ENTITY pcurve
  SUBTYPE OF (curve);
    basis_surface      : surface;
    reference_to_curve : definitional_representation;
  WHERE
    wr1: (SIZEOF(reference_to_curve\representation.items) = 1);
    wr2: ('TECHNICAL_DATA_PACKAGING.CURVE' IN TYPEOF(reference_to_curve\
             representation.items[1]));
    wr3: (reference_to_curve\representation.items[1]\
             geometric_representation_item.dim = 2);
END_ENTITY; -- pcurve

ENTITY person;
    id            : identifier;
    last_name     : OPTIONAL label;
    first_name    : OPTIONAL label;
    middle_names  : OPTIONAL LIST [1:?] OF label;
    prefix_titles : OPTIONAL LIST [1:?] OF label;
    suffix_titles : OPTIONAL LIST [1:?] OF label;
  WHERE
    wr1: (EXISTS(last_name) OR EXISTS(first_name));
END_ENTITY; -- person

ENTITY person_and_organization;
    the_person       : person;
    the_organization : organization;
  DERIVE
```



```
    name        : label := get_name_value(SELF);
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- person_and_organization

ENTITY person_and_organization_assignment
  ABSTRACT SUPERTYPE;
    assigned_person_and_organization : person_and_organization;
    role                             : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment

ENTITY person_and_organization_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- person_and_organization_role

ENTITY personal_address
  SUBTYPE OF (address);
    people      : SET [1:?] OF person;
    description : OPTIONAL text;
END_ENTITY; -- personal_address

ENTITY placement
  SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
      axis2_placement_3d))
  SUBTYPE OF (geometric_representation_item);
    location : cartesian_point;
END_ENTITY; -- placement

ENTITY planar_box
  SUBTYPE OF (planar_extent);
    placement : axis2_placement;
END_ENTITY; -- planar_box

ENTITY planar_extent
  SUBTYPE OF (geometric_representation_item);
    size_in_x : length_measure;
    size_in_y : length_measure;
END_ENTITY; -- planar_extent

ENTITY plane
  SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

ENTITY plane_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.PLANE_ANGLE_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
END_ENTITY; -- plane_angle_measure_with_unit
```

```
ENTITY plane_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
             named_unit.dimensions.mass_exponent = 0) AND (SELF\
             named_unit.dimensions.time_exponent = 0) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
             .dimensions.amount_of_substance_exponent = 0) AND (SELF\
             named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- plane_angle_unit

ENTITY point
  SUPERTYPE OF (ONEOF (cartesian_point,point_on_curve,point_on_surface,
      point_replica,degenerate_pcurve))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point

ENTITY point_on_curve
  SUBTYPE OF (point);
    basis_curve     : curve;
    point_parameter : parameter_value;
END_ENTITY; -- point_on_curve

ENTITY point_on_surface
  SUBTYPE OF (point);
    basis_surface     : surface;
    point_parameter_u : parameter_value;
    point_parameter_v : parameter_value;
END_ENTITY; -- point_on_surface

ENTITY point_replica
  SUBTYPE OF (point);
    parent_pt      : point;
    transformation : cartesian_transformation_operator;
  WHERE
    wr1: (transformation.dim = parent_pt.dim);
    wr2: acyclic_point_replica(SELF,parent_pt);
END_ENTITY; -- point_replica

ENTITY point_style;
    name          : label;
    marker        : marker_select;
    marker_size   : size_select;
    marker_colour : colour;
END_ENTITY; -- point_style

ENTITY poly_loop
  SUBTYPE OF (loop, geometric_representation_item);
    polygon : LIST [3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- poly_loop

ENTITY polyline
  SUBTYPE OF (bounded_curve);
    points : LIST [2:?] OF cartesian_point;
END_ENTITY; -- polyline
```



```
  ENTITY pre_defined_character_glyph
    SUBTYPE OF (pre_defined_item);
  END_ENTITY; -- pre_defined_character_glyph

  ENTITY pre_defined_colour
    SUBTYPE OF (pre_defined_item, colour);
  END_ENTITY; -- pre_defined_colour

  ENTITY pre_defined_curve_font
    SUBTYPE OF (pre_defined_item);
  END_ENTITY; -- pre_defined_curve_font

  ENTITY pre_defined_item;
      name : label;
  END_ENTITY; -- pre_defined_item

  ENTITY pre_defined_text_font
    SUBTYPE OF (pre_defined_item);
  END_ENTITY; -- pre_defined_text_font

  ENTITY precision_qualifier;
      precision_value : INTEGER;
  END_ENTITY; -- precision_qualifier

  ENTITY presentation_area
    SUBTYPE OF (presentation_representation);
    WHERE
      wr1: ((SIZEOF(QUERY ( ais <* USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.'
                + 'AREA_IN_SET.AREA') | (SIZEOF(USEDIN(ais,
                'TECHNICAL_DATA_PACKAGING.' + 'PRESENTATION_SIZE.UNIT')) =
1) ))
                > 0) OR (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
                'PRESENTATION_SIZE.UNIT')) = 1));
  END_ENTITY; -- presentation_area

  ENTITY presentation_representation
    SUBTYPE OF (representation);
    WHERE
      wr1: (SELF\representation.context_of_items\
                geometric_representation_context.coordinate_space_dimension
                = 2);
      wr2: ('TECHNICAL_DATA_PACKAGING.GEOMETRIC_REPRESENTATION_CONTEXT' IN
                 TYPEOF(SELF\representation.context_of_items));
  END_ENTITY; -- presentation_representation

  ENTITY presentation_set;
    INVERSE
      areas : SET [1:?] OF area_in_set FOR in_set;
  END_ENTITY; -- presentation_set

  ENTITY presentation_size;
      unit : presentation_size_assignment_select;
      size : planar_box;
    WHERE
      wr1: ((('TECHNICAL_DATA_PACKAGING.PRESENTATION_REPRESENTATION' IN
                TYPEOF(SELF.unit)) AND item_in_context(SELF.size,SELF.unit\
                representation.context_of_items)) OR ((
                'TECHNICAL_DATA_PACKAGING.AREA_IN_SET' IN TYPEOF(SELF.unit))
                AND (SIZEOF(QUERY ( ais <* SELF.unit\area_in_set.in_set.
                areas | (NOT item_in_context(SELF.size,ais.area\
```



```
              representation.context_of_items)) )) = 0)));
  END_ENTITY; -- presentation_size

  ENTITY presentation_style_assignment;
      styles : SET [1:?] OF presentation_style_select;
    WHERE
      wr1: (SIZEOF(QUERY ( style1 <* SELF.styles | (NOT (SIZEOF(
              QUERY ( style2 <* (SELF.styles - style1) | (NOT ((TYPEOF(
              style1) <> TYPEOF(style2)) OR (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.' + 'SURFACE_STYLE_USAGE',
              'TECHNICAL_DATA_PACKAGING.' + 'EXTERNALLY_DEFINED_STYLE'] *
              TYPEOF(style1)) = 1))) )) = 0)) )) = 0);
      wr2: (SIZEOF(QUERY ( style1 <* SELF.styles | (
              'TECHNICAL_DATA_PACKAGING.SURFACE_STYLE_USAGE' IN TYPEOF(
              style1)) )) <= 2);
  END_ENTITY; -- presentation_style_assignment

  ENTITY presentation_style_by_context
    SUBTYPE OF (presentation_style_assignment);
      style_context : style_context_select;
  END_ENTITY; -- presentation_style_by_context

  ENTITY presentation_view
    SUBTYPE OF (presentation_representation);
  END_ENTITY; -- presentation_view

  ENTITY presented_item
    ABSTRACT SUPERTYPE;
  END_ENTITY; -- presented_item

  ENTITY presented_item_representation;
      presentation : presentation_representation_select;
      item         : presented_item;
  END_ENTITY; -- presented_item_representation

  ENTITY process_product_association;
      name            : label;
      description     : text;
      defined_product : characterized_product_definition;
      process         : product_definition_process;
  END_ENTITY; -- process_product_association

  ENTITY product;
      id                : identifier;
      name              : label;
      description       : OPTIONAL text;
      frame_of_reference : SET [1:?] OF product_context;
  END_ENTITY; -- product

  ENTITY product_category;
      name        : label;
      description : OPTIONAL text;
    DERIVE
      id : identifier := get_id_value(SELF);
    WHERE
      wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
  END_ENTITY; -- product_category
```



```
ENTITY product_category_relationship;
    name         : label;
    description  : OPTIONAL text;
    category     : product_category;
    sub_category : product_category;
  WHERE
    wr1: acyclic_product_category_relationship(SELF,[SELF.sub_category]);
END_ENTITY; -- product_category_relationship

ENTITY product_concept;
    id             : identifier;
    name           : label;
    description    : OPTIONAL text;
    market_context : product_concept_context;
  UNIQUE
    ur1 : id;
END_ENTITY; -- product_concept

ENTITY product_concept_context
  SUBTYPE OF (application_context_element);
    market_segment_type : label;
END_ENTITY; -- product_concept_context

ENTITY product_context
  SUBTYPE OF (application_context_element);
    discipline_type : label;
END_ENTITY; -- product_context

ENTITY product_definition;
    id                : identifier;
    description       : OPTIONAL text;
    formation         : product_definition_formation;
    frame_of_reference : product_definition_context;
  DERIVE
    name : label := get_name_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- product_definition

ENTITY product_definition_context
  SUBTYPE OF (application_context_element);
    life_cycle_stage : label;
END_ENTITY; -- product_definition_context

ENTITY product_definition_context_association;
    definition         : product_definition;
    frame_of_reference : product_definition_context;
    role               : product_definition_context_role;
END_ENTITY; -- product_definition_context_association

ENTITY product_definition_context_role;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- product_definition_context_role

ENTITY product_definition_effectivity
  SUBTYPE OF (effectivity);
    usage : product_definition_relationship;
```

```
    WHERE
      wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
                'EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) = 0);
  END_ENTITY; -- product_definition_effectivity

  ENTITY product_definition_formation;
      id          : identifier;
      description : OPTIONAL text;
      of_product  : product;
    UNIQUE
      ur1 : id, of_product;
  END_ENTITY; -- product_definition_formation

  ENTITY product_definition_formation_relationship;
      id                                   : identifier;
      name                                 : label;
      description                          : OPTIONAL text;
      relating_product_definition_formation : product_definition_formation;
      related_product_definition_formation  : product_definition_formation;
  END_ENTITY; -- product_definition_formation_relationship

  ENTITY product_definition_formation_with_specified_source
    SUBTYPE OF (product_definition_formation);
      make_or_buy : source;
  END_ENTITY; -- product_definition_formation_with_specified_source

  ENTITY product_definition_occurrence_relationship;
      name             : label;
      description      : OPTIONAL text;
      occurrence       : product_definition;
      occurrence_usage : assembly_component_usage;
    WHERE
      wr1: (occurrence_usage.relating_product_definition :<>: occurrence);
      wr2: (occurrence_usage.related_product_definition :<>: occurrence);
      wr3: (occurrence.formation :=: occurrence_usage.
                related_product_definition.formation);
  END_ENTITY; -- product_definition_occurrence_relationship

  ENTITY product_definition_process
    SUBTYPE OF (action);
      identification : identifier;
    INVERSE
      product_definitions : SET [1:?] OF process_product_association FOR
                                process;
  END_ENTITY; -- product_definition_process

  ENTITY product_definition_relationship;
      id                          : identifier;
      name                        : label;
      description                 : OPTIONAL text;
      relating_product_definition : product_definition;
      related_product_definition  : product_definition;
  END_ENTITY; -- product_definition_relationship

  ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
    UNIQUE
      ur1 : definition;
    WHERE
```



```
    wr1: (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.CHARACTERIZED_PRODUCT_DEFINITION',
              'TECHNICAL_DATA_PACKAGING.CHARACTERIZED_OBJECT'] * TYPEOF(
              SELF\property_definition.definition)) > 0);
END_ENTITY; -- product_definition_shape

ENTITY product_definition_substitute;
    description           : OPTIONAL text;
    context_relationship  : product_definition_relationship;
    substitute_definition : product_definition;
  DERIVE
    name : label := get_name_value(SELF);
  WHERE
    wr1: (context_relationship.related_product_definition :<>:
              substitute_definition);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- product_definition_substitute

ENTITY product_definition_usage
  SUPERTYPE OF (ONEOF (make_from_usage_option,assembly_component_usage))
  SUBTYPE OF (product_definition_relationship);
  UNIQUE
    ur1 : id, relating_product_definition, related_product_definition;
  WHERE
    wr1: acyclic_product_definition_relationship(SELF,[SELF\
              product_definition_relationship.related_product_definition],
              'TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_USAGE');
END_ENTITY; -- product_definition_usage

ENTITY product_definition_with_associated_documents
  SUBTYPE OF (product_definition);
    documentation_ids : SET [1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents

ENTITY product_related_product_category
  SUBTYPE OF (product_category);
    products : SET [1:?] OF product;
END_ENTITY; -- product_related_product_category

ENTITY product_relationship;
    id               : identifier;
    name             : label;
    description      : OPTIONAL text;
    relating_product : product;
    related_product  : product;
END_ENTITY; -- product_relationship

ENTITY promissory_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY; -- promissory_usage_occurrence

ENTITY property_definition;
    name        : label;
    description : OPTIONAL text;
    definition  : characterized_definition;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
```

```
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
END_ENTITY; -- property_definition

ENTITY property_definition_relationship;
    name                        : label;
    description                 : text;
    relating_property_definition : property_definition;
    related_property_definition  : property_definition;
END_ENTITY; -- property_definition_relationship

ENTITY property_definition_representation;
    definition          : represented_definition;
    used_representation : representation;
  DERIVE
    description : text := get_description_value(SELF);
    name        : label := get_name_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1);
END_ENTITY; -- property_definition_representation

ENTITY qualified_representation_item
  SUBTYPE OF (representation_item);
    qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    wr1: (SIZEOF(QUERY ( temp <* qualifiers | (
              'TECHNICAL_DATA_PACKAGING.PRECISION_QUALIFIER' IN TYPEOF(
              temp)) )) < 2);
END_ENTITY; -- qualified_representation_item

ENTITY quantified_assembly_component_usage
  SUBTYPE OF (assembly_component_usage);
    quantity : measure_with_unit;
  WHERE
    wr1: ((NOT ('NUMBER' IN TYPEOF(quantity.value_component))) OR (
              quantity.value_component > 0));
END_ENTITY; -- quantified_assembly_component_usage

ENTITY quasi_uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- quasi_uniform_curve

ENTITY quasi_uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.RATIO_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
END_ENTITY; -- ratio_measure_with_unit

ENTITY ratio_unit
  SUBTYPE OF (named_unit);
  WHERE
```



```
    wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- ratio_unit

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
    weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
                 list_to_array(weights_data,0,
                 upper_index_on_control_points);
  WHERE
    wr1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.
              control_points_list));
    wr2: curve_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_curve

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
    weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
                 make_array_of_array(weights_data,0,u_upper,0,v_upper);
  WHERE
    wr1: ((SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.
              control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(
              SELF\b_spline_surface.control_points_list[1])));
    wr2: surface_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_surface

ENTITY rectangular_composite_surface
  SUBTYPE OF (bounded_surface);
    segments : LIST [1:?] OF LIST [1:?] OF surface_patch;
  DERIVE
    n_u : INTEGER := SIZEOF(segments);
    n_v : INTEGER := SIZEOF(segments[1]);
  WHERE
    wr1: (SIZEOF(QUERY ( s <* segments | (n_v <> SIZEOF(s)) )) = 0);
    wr2: constraints_rectangular_composite_surface(SELF);
END_ENTITY; -- rectangular_composite_surface

ENTITY rectangular_trimmed_surface
  SUBTYPE OF (bounded_surface);
    basis_surface : surface;
    u1            : parameter_value;
    u2            : parameter_value;
    v1            : parameter_value;
    v2            : parameter_value;
    usense        : BOOLEAN;
    vsense        : BOOLEAN;
  WHERE
    wr1: (u1 <> u2);
    wr2: (v1 <> v2);
```

```
    wr3: ((('TECHNICAL_DATA_PACKAGING.ELEMENTARY_SURFACE' IN TYPEOF(
            basis_surface)) AND (NOT ('TECHNICAL_DATA_PACKAGING.PLANE'
            IN TYPEOF(basis_surface)))) OR (
            'TECHNICAL_DATA_PACKAGING.SURFACE_OF_REVOLUTION' IN TYPEOF(
            basis_surface)) OR (usense = (u2 > u1)));
    wr4: (('TECHNICAL_DATA_PACKAGING.SPHERICAL_SURFACE' IN TYPEOF(
            basis_surface)) OR (
            'TECHNICAL_DATA_PACKAGING.TOROIDAL_SURFACE' IN TYPEOF(
            basis_surface)) OR (vsense = (v2 > v1)));
END_ENTITY; -- rectangular_trimmed_surface

ENTITY relative_event_occurrence
  SUBTYPE OF (event_occurrence);
    base_event : event_occurrence;
    offset     : time_measure_with_unit;
END_ENTITY; -- relative_event_occurrence

ENTITY reparametrised_composite_curve_segment
  SUBTYPE OF (composite_curve_segment);
    param_length : parameter_value;
  WHERE
    wr1: (param_length > 0);
END_ENTITY; -- reparametrised_composite_curve_segment

ENTITY representation;
    name             : label;
    items            : SET [1:?] OF representation_item;
    context_of_items : representation_context;
  DERIVE
    id          : identifier := get_id_value(SELF);
    description : text := get_description_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
    wr2: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1);
END_ENTITY; -- representation

ENTITY representation_context;
    context_identifier : identifier;
    context_type       : text;
  INVERSE
    representations_in_context : SET [1:?] OF representation FOR
                                    context_of_items;
END_ENTITY; -- representation_context

ENTITY representation_item;
    name : label;
  WHERE
    wr1: (SIZEOF(using_representations(SELF)) > 0);
END_ENTITY; -- representation_item

ENTITY representation_item_relationship;
    name                        : label;
    description                 : OPTIONAL text;
    relating_representation_item : representation_item;
    related_representation_item  : representation_item;
END_ENTITY; -- representation_item_relationship
```



```
ENTITY representation_map;
    mapping_origin        : representation_item;
    mapped_representation : representation;
  INVERSE
    map_usage : SET [1:?] OF mapped_item FOR mapping_source;
  WHERE
    wr1: item_in_context(SELF.mapping_origin,SELF.mapped_representation.
              context_of_items);
END_ENTITY; -- representation_map

ENTITY representation_relationship;
    name        : label;
    description : OPTIONAL text;
    rep_1       : representation;
    rep_2       : representation;
END_ENTITY; -- representation_relationship

ENTITY representation_relationship_with_transformation
  SUBTYPE OF (representation_relationship);
    transformation_operator : transformation;
  WHERE
    wr1: (SELF\representation_relationship.rep_1.context_of_items :<>:
              SELF\representation_relationship.rep_2.context_of_items);
END_ENTITY; -- representation_relationship_with_transformation

ENTITY revolved_face_solid
  SUBTYPE OF (swept_face_solid);
    axis  : axis1_placement;
    angle : plane_angle_measure;
  DERIVE
    axis_line : line := representation_item('') ||
                    geometric_representation_item() || curve() || line(
                    axis.location,representation_item('') ||
                    geometric_representation_item() || vector(axis.z,1));
END_ENTITY; -- revolved_face_solid

ENTITY right_angular_wedge
  SUBTYPE OF (geometric_representation_item);
    position : axis2_placement_3d;
    x        : positive_length_measure;
    y        : positive_length_measure;
    z        : positive_length_measure;
    ltx      : length_measure;
  WHERE
    wr1: ((0 <= ltx) AND (ltx < x));
END_ENTITY; -- right_angular_wedge

ENTITY right_circular_cone
  SUBTYPE OF (geometric_representation_item);
    position   : axis1_placement;
    height     : positive_length_measure;
    radius     : length_measure;
    semi_angle : plane_angle_measure;
  WHERE
    wr1: (radius >= 0);
END_ENTITY; -- right_circular_cone

ENTITY right_circular_cylinder
  SUBTYPE OF (geometric_representation_item);
```

```
    position : axis1_placement;
    height   : positive_length_measure;
    radius   : positive_length_measure;
END_ENTITY; -- right_circular_cylinder

ENTITY role_association;
    role           : object_role;
    item_with_role : role_select;
END_ENTITY; -- role_association

ENTITY seam_curve
  SUBTYPE OF (surface_curve);
  WHERE
    wr1: (SIZEOF(SELF\surface_curve.associated_geometry) = 2);
    wr2: (associated_surface(SELF\surface_curve.associated_geometry[1])
            = associated_surface(SELF\surface_curve.associated_geometry[
            2]));
    wr3: ('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(SELF\surface_curve
            .associated_geometry[1]));
    wr4: ('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(SELF\surface_curve
            .associated_geometry[2]));
END_ENTITY; -- seam_curve

ENTITY security_classification;
    name           : label;
    purpose        : text;
    security_level : security_classification_level;
END_ENTITY; -- security_classification

ENTITY security_classification_assignment
  ABSTRACT SUPERTYPE;
    assigned_security_classification : security_classification;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1);
END_ENTITY; -- security_classification_assignment

ENTITY security_classification_level;
    name : label;
END_ENTITY; -- security_classification_level

ENTITY serial_numbered_effectivity
  SUBTYPE OF (effectivity);
    effectivity_start_id : identifier;
    effectivity_end_id   : OPTIONAL identifier;
END_ENTITY; -- serial_numbered_effectivity

ENTITY shape_aspect;
    name                 : label;
    description          : OPTIONAL text;
    of_shape             : product_definition_shape;
    product_definitional : LOGICAL;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
            'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
```



```
  END_ENTITY; -- shape_aspect

  ENTITY shape_aspect_relationship;
      name                  : label;
      description           : OPTIONAL text;
      relating_shape_aspect : shape_aspect;
      related_shape_aspect  : shape_aspect;
    DERIVE
      id : identifier := get_id_value(SELF);
    WHERE
      wr1: (SIZEOF(USEDIN(SELF,'TECHNICAL_DATA_PACKAGING.' +
                'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1);
  END_ENTITY; -- shape_aspect_relationship

  ENTITY shape_definition_representation
    SUBTYPE OF (property_definition_representation);
    WHERE
      wr1: (('TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(SELF.definition)) OR (
                'TECHNICAL_DATA_PACKAGING.SHAPE_DEFINITION' IN TYPEOF(SELF.
                definition.definition)));
      wr2: ('TECHNICAL_DATA_PACKAGING.SHAPE_REPRESENTATION' IN TYPEOF(SELF
                .used_representation));
  END_ENTITY; -- shape_definition_representation

  ENTITY shape_representation
    SUBTYPE OF (representation);
  END_ENTITY; -- shape_representation

  ENTITY shape_representation_relationship
    SUBTYPE OF (representation_relationship);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.SHAPE_REPRESENTATION' IN (TYPEOF(
                SELF\representation_relationship.rep_1) + TYPEOF(SELF\
                representation_relationship.rep_2)));
  END_ENTITY; -- shape_representation_relationship

  ENTITY shell_based_surface_model
    SUBTYPE OF (geometric_representation_item);
      sbsm_boundary : SET [1:?] OF shell;
    WHERE
      wr1: constraints_geometry_shell_based_surface_model(SELF);
  END_ENTITY; -- shell_based_surface_model

  ENTITY shell_based_wireframe_model
    SUBTYPE OF (geometric_representation_item);
      sbwm_boundary : SET [1:?] OF shell;
    WHERE
      wr1: constraints_geometry_shell_based_wireframe_model(SELF);
  END_ENTITY; -- shell_based_wireframe_model

  ENTITY shell_based_wireframe_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM',
                'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT_3D'] *
TYPEOF(it))
```

```
                = 1)) )) = 0);
      wr2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL',
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM'] * TYPEOF(it)) = 1)
))
                >= 1);
      wr3 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
                edge_list | (NOT ('TECHNICAL_DATA_PACKAGING.EDGE_CURVE' IN
                TYPEOF(el.edge_element))) )) = 0)) )) = 0)) )) = 0)) )) =
0);
      wr4 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( pline_el <*
                QUERY ( el <* eloop\path.edge_list | (
                'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(el.
                edge_element\edge_curve.edge_geometry)) ) | (NOT (SIZEOF(
                pline_el.edge_element\edge_curve.edge_geometry\polyline.
                points) > 2)) )) = 0)) )) = 0)) )) = 0)) )) = 0);
      wr5 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
                edge_list | (NOT valid_wireframe_edge_curve(el.edge_element
                \edge_curve.edge_geometry)) )) = 0)) )) = 0)) )) = 0)) )) =
                 0);
      wr6 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
                edge_list | (NOT (('TECHNICAL_DATA_PACKAGING.VERTEX_POINT'
                IN TYPEOF(el.edge_element.edge_start)) AND (
                'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(el.
                edge_element.edge_end)))) )) = 0)) )) = 0)) )) = 0)) )) =
0);
      wr7 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
```



```
                NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.EDGE_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
                edge_list | (NOT (valid_wireframe_vertex_point(el.
                edge_element.edge_start\vertex_point.vertex_geometry) AND
                valid_wireframe_vertex_point(el.edge_element.edge_end\
                vertex_point.vertex_geometry))) )) = 0)) )) = 0)) )) = 0))
))
                = 0);
      wr8 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.VERTEX_LOOP'
                IN TYPEOF(wsb)) ) | (NOT (
                'TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(vloop\
                vertex_loop.loop_vertex))) )) = 0)) )) = 0)) )) = 0);
      wr9 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(sb)) ) | (
                NOT (SIZEOF(QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
                wire_shell_extent | ('TECHNICAL_DATA_PACKAGING.VERTEX_LOOP'
                IN TYPEOF(wsb)) ) | (NOT valid_wireframe_vertex_point(vloop
                \vertex_loop.loop_vertex\vertex_point.vertex_geometry)) ))
                = 0)) )) = 0)) )) = 0);
      wr10: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_SHELL' IN TYPEOF(sb)) ) |
                (NOT ('TECHNICAL_DATA_PACKAGING.VERTEX_POINT' IN TYPEOF(vs\
                vertex_shell.vertex_shell_extent.loop_vertex))) )) = 0)) ))
                = 0);
      wr11: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.SHELL_BASED_WIREFRAME_MODEL' IN
                TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <* QUERY ( sb <*
                sbwm\shell_based_wireframe_model.sbwm_boundary | (
                'TECHNICAL_DATA_PACKAGING.VERTEX_SHELL' IN TYPEOF(sb)) ) |
                (NOT valid_wireframe_vertex_point(vs\vertex_shell.
                vertex_shell_extent.loop_vertex\vertex_point.
                vertex_geometry)) )) = 0)) )) = 0);
      wr12: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
                'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(it)) ) | (
                NOT (('TECHNICAL_DATA_PACKAGING.' +
                'SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
                mapped_item.mapping_source.mapped_representation))) )) =
0);
      wr13: (SELF.context_of_items\geometric_representation_context.
                coordinate_space_dimension = 3);
  END_ENTITY; -- shell_based_wireframe_shape_representation

  ENTITY si_unit
    SUBTYPE OF (named_unit);
      prefix : OPTIONAL si_prefix;
      name   : si_unit_name;
```



```
    DERIVE
      SELF\named_unit.dimensions : dimensional_exponents :=
                                       dimensions_for_si_unit(name);
  END_ENTITY; -- si_unit

  ENTITY solid_angle_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: ('TECHNICAL_DATA_PACKAGING.SOLID_ANGLE_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
  END_ENTITY; -- solid_angle_measure_with_unit

  ENTITY solid_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
  END_ENTITY; -- solid_angle_unit

  ENTITY solid_model
    SUPERTYPE OF (ONEOF (csg_solid,manifold_solid_brep,swept_face_solid,
        solid_replica))
    SUBTYPE OF (geometric_representation_item);
  END_ENTITY; -- solid_model

  ENTITY solid_replica
    SUBTYPE OF (solid_model);
      parent_solid   : solid_model;
      transformation : cartesian_transformation_operator_3d;
    WHERE
      wr1: acyclic_solid_replica(SELF,parent_solid);
      wr2: (parent_solid\geometric_representation_item.dim = 3);
  END_ENTITY; -- solid_replica

  ENTITY specified_higher_usage_occurrence
    SUBTYPE OF (assembly_component_usage);
      upper_usage : assembly_component_usage;
      next_usage  : next_assembly_usage_occurrence;
    UNIQUE
      ur1 : upper_usage, next_usage;
    WHERE
      wr1: (SELF :<>: upper_usage);
      wr2: (SELF\product_definition_relationship.
              relating_product_definition :=: upper_usage.
              relating_product_definition);
      wr3: (SELF\product_definition_relationship.
              related_product_definition :=: next_usage.
              related_product_definition);
      wr4: ((upper_usage.related_product_definition :=: next_usage.
              relating_product_definition) OR (SIZEOF(QUERY ( pdr <*
              USEDIN(upper_usage.related_product_definition,
              'TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_RELATIONSHIP.'
              + 'RELATED_PRODUCT_DEFINITION') | (pdr.
```



```
            relating_product_definition :=: next_usage.
            relating_product_definition) )) = 1));
    wr5: (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.NEXT_ASSEMBLY_USAGE_OCCURRENCE'
           ,'TECHNICAL_DATA_PACKAGING' +
            '.SPECIFIED_HIGHER_USAGE_OCCURRENCE']
            * TYPEOF(upper_usage)) = 1);
END_ENTITY; -- specified_higher_usage_occurrence

ENTITY sphere
  SUBTYPE OF (geometric_representation_item);
    radius : positive_length_measure;
    centre : point;
END_ENTITY; -- sphere

ENTITY spherical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- spherical_surface

ENTITY styled_item
  SUBTYPE OF (representation_item);
    styles : SET [1:?] OF presentation_style_assignment;
    item   : representation_item;
  WHERE
    wr1: ((SIZEOF(SELF.styles) = 1) XOR (SIZEOF(QUERY ( pres_style <*
            SELF.styles | (NOT (('TECHNICAL_DATA_PACKAGING.' +
            'PRESENTATION_STYLE_BY_CONTEXT') IN TYPEOF(pres_style))) ))
            = 0));
END_ENTITY; -- styled_item

ENTITY surface
  SUPERTYPE OF (ONEOF (elementary_surface,swept_surface,bounded_surface,
      offset_surface,surface_replica))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

ENTITY surface_curve
  SUPERTYPE OF (ONEOF (intersection_curve,seam_curve) ANDOR
      bounded_surface_curve)
  SUBTYPE OF (curve);
    curve_3d              : curve;
    associated_geometry   : LIST [1:2] OF pcurve_or_surface;
    master_representation : preferred_surface_curve_representation;
  DERIVE
    basis_surface : SET [1:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: (curve_3d.dim = 3);
    wr2: (('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(
            associated_geometry[1])) OR (master_representation <>
            pcurve_s1));
    wr3: (('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(
            associated_geometry[2])) OR (master_representation <>
            pcurve_s2));
    wr4: (NOT ('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(curve_3d)));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
```

```
    extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution
  SUBTYPE OF (swept_surface);
    axis_position : axis1_placement;
  DERIVE
    axis_line : line := representation_item('') ||
                  geometric_representation_item() || curve() || line(
                  axis_position.location,representation_item('') ||
                  geometric_representation_item() || vector(
                  axis_position.z,1));
END_ENTITY; -- surface_of_revolution

ENTITY surface_patch
  SUBTYPE OF (founded_item);
    parent_surface : bounded_surface;
    u_transition   : transition_code;
    v_transition   : transition_code;
    u_sense        : BOOLEAN;
    v_sense        : BOOLEAN;
  INVERSE
    using_surfaces : BAG [1:?] OF rectangular_composite_surface FOR
                        segments;
  WHERE
    wr1: (NOT ('TECHNICAL_DATA_PACKAGING.CURVE_BOUNDED_SURFACE' IN
              TYPEOF(parent_surface)));
END_ENTITY; -- surface_patch

ENTITY surface_rendering_properties;
    rendered_colour : colour;
END_ENTITY; -- surface_rendering_properties

ENTITY surface_replica
  SUBTYPE OF (surface);
    parent_surface : surface;
    transformation : cartesian_transformation_operator_3d;
  WHERE
    wr1: acyclic_surface_replica(SELF,parent_surface);
END_ENTITY; -- surface_replica

ENTITY surface_side_style;
    name   : label;
    styles : SET [1:7] OF surface_style_element_select;
  WHERE
    wr1: (SIZEOF(QUERY ( style1 <* SELF.styles | (SIZEOF(
              QUERY ( style2 <* (SELF.styles - style1) | (TYPEOF(style1) =
               TYPEOF(style2)) )) > 0) )) = 0);
END_ENTITY; -- surface_side_style

ENTITY surface_style_boundary;
    style_of_boundary : curve_or_render;
END_ENTITY; -- surface_style_boundary

ENTITY surface_style_control_grid;
    style_of_control_grid : curve_or_render;
END_ENTITY; -- surface_style_control_grid
```



```
ENTITY surface_style_fill_area;
    fill_area : fill_area_style;
END_ENTITY; -- surface_style_fill_area

ENTITY surface_style_parameter_line;
    style_of_parameter_lines : curve_or_render;
    direction_counts         : SET [1:2] OF direction_count_select;
  WHERE
    wr1: ((HIINDEX(SELF.direction_counts) = 1) XOR (TYPEOF(SELF.
             direction_counts[1]) <> TYPEOF(SELF.direction_counts[2])));
END_ENTITY; -- surface_style_parameter_line

ENTITY surface_style_reflectance_ambient;
    ambient_reflectance : REAL;
END_ENTITY; -- surface_style_reflectance_ambient

ENTITY surface_style_reflectance_ambient_diffuse
  SUBTYPE OF (surface_style_reflectance_ambient);
    diffuse_reflectance : REAL;
END_ENTITY; -- surface_style_reflectance_ambient_diffuse

ENTITY surface_style_reflectance_ambient_diffuse_specular
  SUBTYPE OF (surface_style_reflectance_ambient_diffuse);
    specular_reflectance : REAL;
    specular_exponent    : REAL;
    specular_colour      : colour;
END_ENTITY; -- surface_style_reflectance_ambient_diffuse_specular

ENTITY surface_style_rendering;
    rendering_method : shading_surface_method;
    surface_colour   : colour;
END_ENTITY; -- surface_style_rendering

ENTITY surface_style_rendering_with_properties
  SUBTYPE OF (surface_style_rendering);
    properties : SET [1:2] OF rendering_properties_select;
  WHERE
    wr1: ((HIINDEX(SELF.properties) = 1) XOR (TYPEOF(SELF.properties[1])
             <> TYPEOF(SELF.properties[2])));
END_ENTITY; -- surface_style_rendering_with_properties

ENTITY surface_style_segmentation_curve;
    style_of_segmentation_curve : curve_or_render;
END_ENTITY; -- surface_style_segmentation_curve

ENTITY surface_style_silhouette;
    style_of_silhouette : curve_or_render;
END_ENTITY; -- surface_style_silhouette

ENTITY surface_style_transparent;
    transparency : REAL;
  WHERE
    wr1: ((0 <= transparency) AND (transparency <= 1));
END_ENTITY; -- surface_style_transparent

ENTITY surface_style_usage;
    side  : surface_side;
    style : surface_side_style_select;
END_ENTITY; -- surface_style_usage
```

```
ENTITY swept_face_solid
  SUPERTYPE OF (ONEOF (extruded_face_solid,revolved_face_solid))
  SUBTYPE OF (solid_model);
    swept_face : face_surface;
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.PLANE' IN TYPEOF(swept_face.
              face_geometry));
END_ENTITY; -- swept_face_solid

ENTITY swept_surface
  SUPERTYPE OF (ONEOF
   (surface_of_linear_extrusion,surface_of_revolution))
  SUBTYPE OF (surface);
    swept_curve : curve;
END_ENTITY; -- swept_surface

ENTITY symbol_representation
  SUBTYPE OF (representation);
END_ENTITY; -- symbol_representation

ENTITY text_literal
  SUBTYPE OF (geometric_representation_item);
    literal   : presentable_text;
    placement : axis2_placement;
    alignment : text_alignment;
    path      : text_path;
    font      : font_select;
END_ENTITY; -- text_literal

ENTITY text_string_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: (SIZEOF(QUERY ( item <* SELF\representation.items | (SIZEOF([
              'TECHNICAL_DATA_PACKAGING.TEXT_LITERAL',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT_CHARACTER',
              'TECHNICAL_DATA_PACKAGING.DEFINED_CHARACTER_GLYPH',
              'TECHNICAL_DATA_PACKAGING.COMPOSITE_TEXT',
              'TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT'] * TYPEOF(item))
              = 0) )) = 0);
    wr2: (SIZEOF(QUERY ( item <* SELF\representation.items | (NOT (
              SIZEOF(['TECHNICAL_DATA_PACKAGING.TEXT_LITERAL',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT',
              'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT_CHARACTER',
              'TECHNICAL_DATA_PACKAGING.DEFINED_CHARACTER_GLYPH',
              'TECHNICAL_DATA_PACKAGING.COMPOSITE_TEXT'] * TYPEOF(item)) =
               0)) )) >= 1);
    wr3: (SIZEOF(QUERY ( a2p <* QUERY ( item <* SELF\representation.
              items | ('TECHNICAL_DATA_PACKAGING.AXIS2_PLACEMENT' IN
              TYPEOF(item)) ) | (NOT ((SIZEOF(QUERY ( at <*
              QUERY ( item <* SELF\representation.items | ((
              'TECHNICAL_DATA_PACKAGING.' + 'ANNOTATION_TEXT') IN TYPEOF(
              item)) ) | (at\mapped_item.mapping_target :=: a2p) )) >= 1)
              OR (SIZEOF(QUERY ( atc <* QUERY ( item <* SELF\
              representation.items | (('TECHNICAL_DATA_PACKAGING.' +
              'ANNOTATION_TEXT_CHARACTER') IN TYPEOF(item)) ) | (atc\
              mapped_item.mapping_target :=: a2p) )) >= 1))) )) = 0);
END_ENTITY; -- text_string_representation
```



```
ENTITY text_style;
    name                 : label;
    character_appearance : character_style_select;
END_ENTITY; -- text_style

ENTITY text_style_for_defined_font;
    text_colour : colour;
END_ENTITY; -- text_style_for_defined_font

ENTITY thermodynamic_temperature_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.THERMODYNAMIC_TEMPERATURE_UNIT' IN
              TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- thermodynamic_temperature_measure_with_unit

ENTITY thermodynamic_temperature_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 1) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- thermodynamic_temperature_unit

ENTITY time_interval;
    id          : identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- time_interval

ENTITY time_interval_based_effectivity
  SUBTYPE OF (effectivity);
    effectivity_period : time_interval;
END_ENTITY; -- time_interval_based_effectivity

ENTITY time_interval_relationship;
    name                   : label;
    description            : OPTIONAL text;
    relating_time_interval : time_interval;
    related_time_interval  : time_interval;
END_ENTITY; -- time_interval_relationship

ENTITY time_interval_with_bounds
  SUBTYPE OF (time_interval);
    primary_bound   : OPTIONAL date_time_or_event_occurrence;
    secondary_bound : OPTIONAL date_time_or_event_occurrence;
    duration        : OPTIONAL time_measure_with_unit;
  WHERE
    wr1: (NOT (EXISTS(secondary_bound) AND EXISTS(duration)));
    wr2: (EXISTS(primary_bound) OR EXISTS(secondary_bound));
END_ENTITY; -- time_interval_with_bounds
```

```
ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.TIME_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
END_ENTITY; -- time_measure_with_unit

ENTITY time_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 1) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- time_unit

ENTITY topological_representation_item
  SUPERTYPE OF (ONEOF (vertex,edge,face_bound,face,vertex_shell,
      wire_shell,connected_edge_set,connected_face_set,loop ANDOR path))
  SUBTYPE OF (representation_item);
END_ENTITY; -- topological_representation_item

ENTITY toroidal_surface
  SUBTYPE OF (elementary_surface);
    major_radius : positive_length_measure;
    minor_radius : positive_length_measure;
END_ENTITY; -- toroidal_surface

ENTITY torus
  SUBTYPE OF (geometric_representation_item);
    position     : axis1_placement;
    major_radius : positive_length_measure;
    minor_radius : positive_length_measure;
  WHERE
    wr1: (major_radius > minor_radius);
END_ENTITY; -- torus

ENTITY trimmed_curve
  SUBTYPE OF (bounded_curve);
    basis_curve          : curve;
    trim_1               : SET [1:2] OF trimming_select;
    trim_2               : SET [1:2] OF trimming_select;
    sense_agreement      : BOOLEAN;
    master_representation : trimming_preference;
  WHERE
    wr1: ((HIINDEX(trim_1) = 1) OR (TYPEOF(trim_1[1]) <>
         TYPEOF(trim_1[2])));
    wr2: ((HIINDEX(trim_2) = 1) OR (TYPEOF(trim_2[1]) <>
         TYPEOF(trim_2[2])));
END_ENTITY; -- trimmed_curve

ENTITY type_qualifier;
    name : label;
END_ENTITY; -- type_qualifier
```



```
ENTITY uncertainty_measure_with_unit
  SUBTYPE OF (measure_with_unit);
    name        : label;
    description : OPTIONAL text;
  WHERE
    wr1: valid_measure_value(SELF\measure_with_unit.value_component);
END_ENTITY; -- uncertainty_measure_with_unit

ENTITY uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- uniform_curve

ENTITY uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- uniform_surface

ENTITY value_representation_item
  SUBTYPE OF (representation_item);
    value_component : measure_value;
  WHERE
    wr1: (SIZEOF(QUERY ( rep <* using_representations(SELF) | (NOT (
              'TECHNICAL_DATA_PACKAGING.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN
              TYPEOF(rep.context_of_items))) )) = 0);
END_ENTITY; -- value_representation_item

ENTITY vector
  SUBTYPE OF (geometric_representation_item);
    orientation : direction;
    magnitude   : length_measure;
  WHERE
    wr1: (magnitude >= 0);
END_ENTITY; -- vector

ENTITY versioned_action_request;
    id          : identifier;
    version     : label;
    purpose     : text;
    description : OPTIONAL text;
END_ENTITY; -- versioned_action_request

ENTITY vertex
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- vertex

ENTITY vertex_loop
  SUBTYPE OF (loop);
    loop_vertex : vertex;
END_ENTITY; -- vertex_loop

ENTITY vertex_point
  SUBTYPE OF (vertex, geometric_representation_item);
    vertex_geometry : point;
END_ENTITY; -- vertex_point

ENTITY vertex_shell
  SUBTYPE OF (topological_representation_item);
    vertex_shell_extent : vertex_loop;
END_ENTITY; -- vertex_shell
```

```
ENTITY view_volume
  SUBTYPE OF (founded_item);
    projection_type            : central_or_parallel;
    projection_point           : cartesian_point;
    view_plane_distance        : length_measure;
    front_plane_distance       : length_measure;
    front_plane_clipping       : BOOLEAN;
    back_plane_distance        : length_measure;
    back_plane_clipping        : BOOLEAN;
    view_volume_sides_clipping : BOOLEAN;
    view_window                : planar_box;
END_ENTITY; -- view_volume

ENTITY volume_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('TECHNICAL_DATA_PACKAGING.VOLUME_UNIT' IN TYPEOF(SELF\
              measure_with_unit.unit_component));
END_ENTITY; -- volume_measure_with_unit

ENTITY volume_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 3) AND (SELF\
              named_unit.dimensions.mass_exponent = 0) AND (SELF\
              named_unit.dimensions.time_exponent = 0) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
              .dimensions.amount_of_substance_exponent = 0) AND (SELF\
              named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- volume_unit

ENTITY wire_shell
  SUBTYPE OF (topological_representation_item);
    wire_shell_extent : SET [1:?] OF loop;
  WHERE
    wr1: (NOT mixed_loop_type_set(wire_shell_extent));
END_ENTITY; -- wire_shell

RULE access_file_requires_external_identification FOR (document_file);

  LOCAL
    df     : SET OF document_file;
    aeia   : BAG OF applied_external_identification_assignment;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  df := document_file;
  REPEAT i := 1 TO SIZEOF(df) BY 1;
    IF df[i]\document.description = 'access data file' THEN
      aeia := USEDIN(df[i],'TECHNICAL_DATA_PACKAGING.' +
          'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS');
      IF SIZEOF(aeia) < 1 THEN
        result := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;

WHERE
```



```
  wr1: result;

END_RULE; -- access_file_requires_external_identification

RULE applied_document_usage_constraint_assignment_restriction FOR (
           document_file, externally_defined_symbol_and_placement);

  LOCAL
    exds   : SET OF externally_defined_symbol_and_placement;
    aduca  : BAG OF applied_document_usage_constraint_assignment;
    df     : SET OF document_file;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  df := document_file;
  REPEAT i := 1 TO SIZEOF(df) BY 1;
    aduca := USEDIN(df[i],'TECHNICAL_DATA_PACKAGING.' +
        'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS');
    IF SIZEOF(aduca) > 0 THEN
      result := FALSE;
    END_IF;
  END_REPEAT;
  exds := externally_defined_symbol_and_placement;
  REPEAT i := 1 TO SIZEOF(exds) BY 1;
    aduca := USEDIN(exds[i],'TECHNICAL_DATA_PACKAGING.' +
        'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS');
    IF SIZEOF(aduca) > 0 THEN
      result := FALSE;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- applied_document_usage_constraint_assignment_restriction

RULE approval_requires_approval_date_time FOR (approval,
           approval_date_time);

WHERE
  wr1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( adt <*
           approval_date_time | (app :=: adt.dated_approval) )) = 1)) ))
           = 0);

END_RULE; -- approval_requires_approval_date_time

RULE approval_requires_approval_person_organization FOR (approval,
           approval_person_organization);

WHERE
  wr1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( apo <*
           approval_person_organization | (app :=: apo.
           authorized_approval) )) >= 1)) )) = 0);

END_RULE; -- approval_requires_approval_person_organization

RULE change_identification_restricts_executed_action FOR (
           executed_action);

  LOCAL
```

```
      acta   : BAG OF action_assignment;
      found  : BOOLEAN := FALSE;
      exeact : SET OF executed_action;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    exeact := executed_action;
    REPEAT i := 1 TO SIZEOF(exeact) BY 1;
      IF exeact[i].description = 'change identification' THEN
        acta := USEDIN(exeact[i],
            'TECHNICAL_DATA_PACKAGING.ACTION_ASSIGNMENT.ASSIGNED_ACTION');
        found := FALSE;
        REPEAT j := 1 TO SIZEOF(acta) BY 1;
          IF EXISTS(acta[j].role) THEN
            IF (acta[j].role.name = 'change identification') XOR (acta[j].
                role.name = 'revision history') THEN
              found := TRUE;
            END_IF;
          END_IF;
        END_REPEAT;
        IF NOT found THEN
          result := FALSE;
          ESCAPE;
        END_IF;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- change_identification_restricts_executed_action

  RULE compatible_dimension FOR (cartesian_point, direction,
            representation_context, geometric_representation_context);

  WHERE
    wr1: (SIZEOF(QUERY ( x <* cartesian_point | (SIZEOF(QUERY ( y <*
            geometric_representation_context | (item_in_context(x,y) AND (
            HIINDEX(x.coordinates) <> y.coordinate_space_dimension)) )) >
            0) )) = 0);
    wr2: (SIZEOF(QUERY ( x <* direction | (SIZEOF(QUERY ( y <*
            geometric_representation_context | (item_in_context(x,y) AND (
            HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension))
))
            > 0) )) = 0);

  END_RULE; -- compatible_dimension

  RULE data_definition_entry_string_restrict_for_superseded_element FOR (
            product_definition);

    LOCAL
      pdca    : BAG OF product_definition_context_association;
      result1 : BOOLEAN;
      result2 : BOOLEAN;
      found   : BOOLEAN;
      pd      : SET OF product_definition;
      pdf     : product_definition_formation;
      pdc     : product_definition_context;
    END_LOCAL;
```



```
    result1 := TRUE;
    result2 := TRUE;
    pd := product_definition;
    REPEAT i := 1 TO SIZEOF(pd) BY 1;
      found := FALSE;
      pdca := USEDIN(pd[i],'TECHNICAL_DATA_PACKAGING.' +
          'PRODUCT_DEFINITION_CONTEXT_ASSOCIATION.DEFINITION');
      REPEAT j := 1 TO SIZEOF(pdca) BY 1;
        pdc := pdca[j].frame_of_reference;
        IF pdc\application_context_element.name IN ['document version',
            'a representation of a document version',
            'data definition entry','indentured data list entry'] THEN
          found := TRUE;
        END_IF;
      END_REPEAT;
      IF found THEN
        IF SIZEOF(QUERY ( pdr <* USEDIN(pd[i],'TECHNICAL_DATA_PACKAGING.'
            + 'PRODUCT_DEFINITION_RELATIONSHIP.' +
            'RELATED_PRODUCT_DEFINITION') | (pdr.name =
            'superseded element') )) > 1 THEN
          result1 := FALSE;
        END_IF;
        pdf := pd[i].formation;
        IF SIZEOF(QUERY ( pdfr <* USEDIN(pdf,'TECHNICAL_DATA_PACKAGING.' +
             'PRODUCT_DEFINITION_FORMATION_RELATIONSHIP.' +
            'RELATED_PRODUCT_DEFINITION_FORMATION') | (pdfr.name =
            'superseded version') )) > 1 THEN
          result2 := FALSE;
        END_IF;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result1;
    wr2: result2;

  END_RULE; -- data_definition_entry_string_restrict_for_superseded_element

  RULE data_definition_exchange_categorized_as_document FOR (
            product_related_product_category);

    LOCAL
      prds   : SET OF product := [];
      prpc   : SET OF product_related_product_category;
      pc     : SET OF product_category;
      pdf    : BAG OF product_definition_formation;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    prpc := QUERY ( v <* product_related_product_category | (v.name =
        'data definition exchange') );
    REPEAT i := 1 TO SIZEOF(prpc) BY 1;
      prds := prpc[i].products;
      REPEAT j := 1 TO SIZEOF(prds) BY 1;
        pdf := USEDIN(prds[j],'TECHNICAL_DATA_PACKAGING.' +
            'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT');
        IF SIZEOF(pdf) > 0 THEN
          pc := product_category_graph_members(prpc[i]);
          IF SIZEOF(QUERY ( v <* pc | (v.name = 'document') )) < 1 THEN
            result := FALSE;
```

```
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- data_definition_exchange_categorized_as_document

  RULE data_definition_exchange_restrict_to_one_presentation FOR (
            product_related_product_category);

    LOCAL
      prds   : SET OF product := [];
      prpc   : SET OF product_related_product_category;
      pdf    : BAG OF product_definition_formation;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    prpc := product_related_product_category;
    REPEAT i := 1 TO SIZEOF(prpc) BY 1;
      IF prpc[i]\product_category.name = 'data definition exchange' THEN
        prds := prds + prpc[i].products;
      END_IF;
    END_REPEAT;
    REPEAT i := 1 TO SIZEOF(prds) BY 1;
      pdf := USEDIN(prds[i],'TECHNICAL_DATA_PACKAGING.' +
          'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT');
      REPEAT j := 1 TO SIZEOF(pdf) BY 1;
        IF SIZEOF(USEDIN(pdf[j],'TECHNICAL_DATA_PACKAGING.' +
            'APPLIED_PRESENTED_ITEM.ITEMS')) > 1 THEN
          result := FALSE;
        END_IF;
      END_REPEAT;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- data_definition_exchange_restrict_to_one_presentation

  RULE dependent_instantiable_named_unit FOR (named_unit);

    LOCAL
      nu     : SET OF named_unit;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    nu := named_unit;
    REPEAT i := 1 TO SIZEOF(nu) BY 1;
      IF SIZEOF(USEDIN(nu[i],'')) < 1 THEN
        result := FALSE;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- dependent_instantiable_named_unit
```



```
RULE distribution_notice_approval_requires_supporting_data FOR (
          applied_approval_assignment);

  LOCAL
    ap     : approval;
    aca    : BAG OF applied_classification_assignment;
    found  : BOOLEAN;
    aaa    : SET OF applied_approval_assignment;
    apo    : BAG OF approval_person_organization;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  aaa := applied_approval_assignment;
  REPEAT i := 1 TO SIZEOF(aaa) BY 1;
    IF EXISTS(aaa[i].role) THEN
      IF aaa[i].role.name = 'distribution notice' THEN
        ap := aaa[i].assigned_approval;
        found := FALSE;
        IF LENGTH(ap.level) > 0 THEN
          found := TRUE;
        END_IF;
        apo := USEDIN(ap,'TECHNICAL_DATA_PACKAGING.' +
            'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL');
        REPEAT j := 1 TO SIZEOF(apo) BY 1;
          IF apo[j].role.role = 'distribution authorization' THEN
            found := TRUE;
          END_IF;
        END_REPEAT;
        aca := USEDIN(ap,'TECHNICAL_DATA_PACKAGING.' +
            'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS');
        REPEAT j := 1 TO SIZEOF(aca) BY 1;
          IF EXISTS(aca[j]\classification_assignment.role) THEN
            IF (aca[j]\classification_assignment.role.name =
                'distribution notice code') AND (
                'TECHNICAL_DATA_PACKAGING.CLASS' IN TYPEOF(aca[j]\
                classification_assignment.assigned_class)) THEN
              found := TRUE;
            END_IF;
          END_IF;
        END_REPEAT;
        IF NOT found THEN
          result := FALSE;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- distribution_notice_approval_requires_supporting_data

RULE drawing_suffix_number_combination_identification_constraint FOR (
          product_relationship);

  LOCAL
    prpc    : BAG OF product_related_product_category;
    p       : product;
    tresult : BOOLEAN;
    pr      : SET OF product_relationship;
```

```
      result  : BOOLEAN := TRUE;
    END_LOCAL;
    pr := QUERY ( v <* product_relationship | (v.name =
        'drawing suffix number combination') );
    REPEAT i := 1 TO SIZEOF(pr) BY 1;
      p := pr[i].relating_product;
      prpc := USEDIN(p,'TECHNICAL_DATA_PACKAGING.' +
          'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
      REPEAT j := 1 TO SIZEOF(prpc) BY 1;
        IF prpc[j]\product_category.name = 'document' THEN
          result := FALSE;
        END_IF;
      END_REPEAT;
      p := pr[i].related_product;
      prpc := USEDIN(p,'TECHNICAL_DATA_PACKAGING.' +
          'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
      tresult := FALSE;
      REPEAT j := 1 TO SIZEOF(prpc) BY 1;
        IF prpc[j]\product_category.name = 'document' THEN
          tresult := TRUE;
        END_IF;
      END_REPEAT;
      IF NOT tresult THEN
        result := FALSE;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- drawing_suffix_number_combination_identification_constraint

  RULE exchange_reason_existence_required FOR (property_definition);

    LOCAL
      dri    : descriptive_representation_item;
      prpd   : SET OF property_definition;
      repi   : representation;
      result : BOOLEAN := TRUE;
      prdr   : BAG OF property_definition_representation;
    END_LOCAL;
    prpd := property_definition;
    REPEAT i := 1 TO SIZEOF(prpd) BY 1;
      IF prpd[i].name = 'data definition exchange header' THEN
        result := FALSE;
        prdr := USEDIN(prpd[i],'TECHNICAL_DATA_PACKAGING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
        REPEAT j := 1 TO SIZEOF(prdr) BY 1;
          repi := prdr[j].used_representation;
          IF NOT (repi.name = 'exchange reason') THEN
            ESCAPE;
          ELSE
            REPEAT k := 1 TO SIZEOF(repi.items) BY 1;
              IF 'TECHNICAL_DATA_PACKAGING.DESCRIPTIVE_REPRESENTATION_ITEM'
                  IN TYPEOF(repi.items[k]) THEN
                dri := repi.items[k];
                IF (dri.name = 'exchange purpose') XOR (dri.name =
                    'base exchange reason') THEN
                  result := TRUE;
```



```
                ESCAPE;
              END_IF;
            END_IF;
          END_REPEAT;
        END_IF;
      END_REPEAT;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- exchange_reason_existence_required

RULE existence_dependent_drawing_sheet_revision FOR (
           drawing_sheet_revision);

  LOCAL
    pir    : BAG OF presented_item_representation;
    prpc   : BAG OF product_related_product_category;
    dsr    : SET OF drawing_sheet_revision;
    pdf    : SET OF product_definition_formation;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  dsr := drawing_sheet_revision;
  REPEAT i := 1 TO SIZEOF(dsr) BY 1;
    pir := USEDIN(dsr[i],'TECHNICAL_DATA_PACKAGING.' +
        'PRESENTED_ITEM_REPRESENTATION.PRESENTATION');
    IF SIZEOF(pir) < 1 THEN
      result := FALSE;
    END_IF;
    REPEAT j := 1 TO SIZEOF(pir) BY 1;
      pdf := pir[j].item\applied_presented_item.items;
      IF SIZEOF(pdf) < 1 THEN
        result := FALSE;
      END_IF;
      REPEAT k := 1 TO SIZEOF(pdf) BY 1;
        prpc := USEDIN(pdf[k].of_product,'TECHNICAL_DATA_PACKAGING.' +
            'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
        IF SIZEOF(prpc) < 1 THEN
          result := FALSE;
        END_IF;
        REPEAT l := 1 TO SIZEOF(prpc) BY 1;
          IF prpc[l]\product_category.name <> 'sheet' THEN
            result := FALSE;
          END_IF;
        END_REPEAT;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- existence_dependent_drawing_sheet_revision

RULE file_format_restricted_to_one_data_definition_entry FOR (
           product_definition);
```

```
    LOCAL
      propd  : BAG OF property_definition;
      pd     : SET OF product_definition;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    pd := product_definition;
    REPEAT i := 1 TO SIZEOF(pd) BY 1;
      IF (pd[i].frame_of_reference.name = 'document version') XOR (pd[i].
          frame_of_reference.name =
          'a representation of a document version') THEN
        propd := USEDIN(pd[i],'TECHNICAL_DATA_PACKAGING.' +
            'PROPERTY_DEFINITION.DEFINITION');
        IF SIZEOF(QUERY ( p <* propd | (p.name = 'document format') )) > 1
            THEN
          result := FALSE;
        END_IF;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- file_format_restricted_to_one_data_definition_entry

  RULE header_configuration_restricts_property_definition FOR (
            property_definition);

    LOCAL
      found  : BOOLEAN;
      pd     : SET OF property_definition;
      apa    : BAG OF applied_approval_assignment;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    pd := QUERY ( v <* property_definition | (v.name LIKE '* header') );
    REPEAT i := 1 TO SIZEOF(pd) BY 1;
      found := FALSE;
      apa := USEDIN(pd[i],
          'TECHNICAL_DATA_PACKAGING.APPLIED_APPROVAL_ASSIGNMENT.ITEMS');
      REPEAT j := 1 TO SIZEOF(apa) BY 1;
        IF EXISTS(apa[j]\approval_assignment.role) THEN
          IF (apa[j]\approval_assignment.role.name =
              'release authentication') OR (apa[j]\approval_assignment.role
              .name = 'data usage rights') THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF SIZEOF(USEDIN(pd[i],'TECHNICAL_DATA_PACKAGING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) > 0 THEN
        found := TRUE;
      END_IF;
      IF NOT found THEN
        result := FALSE;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;
```



```
END_RULE; -- header_configuration_restricts_property_definition

RULE identification_of_sheet_constraint FOR (
           product_definition_formation);

  LOCAL
    prpc   : BAG OF product_related_product_category;
    found  : BOOLEAN;
    pc     : SET OF product_category;
    pdf    : SET OF product_definition_formation;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  pdf := product_definition_formation;
  REPEAT i := 1 TO SIZEOF(pdf) BY 1;
    prpc := USEDIN(pdf[i].of_product,'TECHNICAL_DATA_PACKAGING.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    REPEAT j := 1 TO SIZEOF(prpc) BY 1;
      IF prpc[j]\product_category.name = 'sheet' THEN
        pc := product_category_graph_members(prpc[j]);
        found := FALSE;
        REPEAT k := 1 TO SIZEOF(pc) BY 1;
          IF pc[k].name = 'document' THEN
            found := TRUE;
          END_IF;
        END_REPEAT;
        IF NOT found THEN
          result := FALSE;
          ESCAPE;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- identification_of_sheet_constraint

RULE indentured_data_list_identification_constraint FOR (
           product_definition_formation);

  LOCAL
    prpc   : BAG OF product_related_product_category;
    pc     : SET OF product_category;
    pdf    : SET OF product_definition_formation;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  pdf := product_definition_formation;
  REPEAT i := 1 TO SIZEOF(pdf) BY 1;
    prpc := USEDIN(pdf[i].of_product,'TECHNICAL_DATA_PACKAGING.' +
     'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
    REPEAT j := 1 TO SIZEOF(prpc) BY 1;
      IF prpc[j]\product_category.name = 'indentured data list' THEN
        pc := product_category_graph_members(prpc[j]);
        result := FALSE;
        REPEAT k := 1 TO SIZEOF(pc) BY 1;
          IF pc[k].name = 'document' THEN
            result := TRUE;
          END_IF;
```

```
          END_REPEAT;
          IF result = FALSE THEN
            ESCAPE;
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- indentured_data_list_identification_constraint

  RULE indentured_level_tag_identification_constraint FOR (
            property_definition_representation);

    LOCAL
      pdr    : SET OF property_definition_representation;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    pdr := property_definition_representation;
    REPEAT i := 1 TO SIZEOF(pdr) BY 1;
      IF pdr[i].used_representation.name = 'indentured level tag' THEN
        IF NOT ((pdr[i].definition.name = 'exchange entry property') XOR (
            pdr[i].definition.name = 'indentured data list entry
property'))
            THEN
          result := FALSE;
        END_IF;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- indentured_level_tag_identification_constraint

  RULE indentured_list_method_identification_constraint FOR (
            applied_document_reference);

    LOCAL
      prpc   : BAG OF product_related_product_category;
      d      : document;
      p      : product;
      dpa    : BAG OF document_product_association;
      adr    : SET OF applied_document_reference;
      pfd    : product_or_formation_or_definition;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    adr := applied_document_reference;
    REPEAT i := 1 TO SIZEOF(adr) BY 1;
      IF EXISTS(adr[i]\document_reference.role) THEN
        IF (adr[i]\document_reference.role.name = 'indentured by document')
            OR (adr[i]\document_reference.role.name = 'indentured by item')
            OR (adr[i]\document_reference.role.name =
            'indentured by item and document') THEN
          d := adr[i]\document_reference.assigned_document;
          dpa := USEDIN(d,'TECHNICAL_DATA_PACKAGING.' +
              'DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT');
```



```
        REPEAT j := 1 TO SIZEOF(dpa) BY 1;
          pfd := dpa[j].related_product;
          IF 'TECHNICAL_DATA_PACKAGING.PRODUCT' IN TYPEOF(pfd) THEN
            p := pfd;
          ELSE
            IF 'TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION'
                IN TYPEOF(pfd) THEN
              p := pfd\product_definition_formation.of_product;
            ELSE
              IF 'TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION' IN
                  TYPEOF(pfd) THEN
                p := pfd\product_definition.formation.of_product;
              END_IF;
            END_IF;
          END_IF;
          result := FALSE;
          prpc := USEDIN(p,'TECHNICAL_DATA_PACKAGING.' +
              'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
          REPEAT k := 1 TO SIZEOF(prpc) BY 1;
            IF (prpc[k].name = 'data definition exchange') OR (prpc[k].
                name = 'indentured data list') THEN
              result := TRUE;
            END_IF;
          END_REPEAT;
        END_REPEAT;
      END_IF;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- indentured_list_method_identification_constraint

RULE item_source_information_identification_constraint FOR (
          property_definition);

  LOCAL
    prpd   : SET OF property_definition;
    result : BOOLEAN := TRUE;
    prdr   : BAG OF property_definition_representation;
  END_LOCAL;
  prpd := property_definition;
  REPEAT i := 1 TO SIZEOF(prpd) BY 1;
    IF prpd[i].name = 'source information' THEN
      result := FALSE;
      prdr := USEDIN(prpd[i],'TECHNICAL_DATA_PACKAGING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
      REPEAT j := 1 TO SIZEOF(prdr) BY 1;
        IF prdr[j].used_representation.name = 'source information type'
            THEN
          result := TRUE;
        END_IF;
      END_REPEAT;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result;
```

```
  END_RULE; -- item_source_information_identification_constraint

  RULE notation_type_identification_constraint FOR (representation);

    LOCAL
      note   : SET OF representation;
      result : BOOLEAN;
    END_LOCAL;
    note := QUERY ( rep <* representation | (rep.name = 'notation') );
    result := FALSE;
    IF SIZEOF(note) < 1 THEN
      result := TRUE;
    END_IF;
    REPEAT i := 1 TO SIZEOF(note) BY 1;
      REPEAT j := 1 TO SIZEOF(note[i].items) BY 1;
        IF 'TECHNICAL_DATA_PACKAGING.DESCRIPTIVE_REPRESENTATION_ITEM' IN
            TYPEOF(note[i].items[j]) THEN
          result := TRUE;
          ESCAPE;
        END_IF;
      END_REPEAT;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- notation_type_identification_constraint

  RULE product_requires_category FOR (product,
            product_related_product_category);

  WHERE
    wr1: (SIZEOF(QUERY ( p <* product |
    (SIZEOF(USEDIN(p,'TECHNICAL_DATA_PACKAGING.'+
            'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS'))
            = 0) )) = 0);

  END_RULE; -- product_requires_category

  RULE product_requires_version FOR (product);

  WHERE
    wr1: (SIZEOF(QUERY ( prod <* product | (SIZEOF(USEDIN(prod,
              'TECHNICAL_DATA_PACKAGING.' +
              'PRODUCT_DEFINITION_FORMATION.' + 'OF_PRODUCT')) = 0) )) = 0);

  END_RULE; -- product_requires_version

  RULE product_version_requires_person_organization FOR (
            product_definition_formation);

    LOCAL
      apos   : BAG OF applied_person_and_organization_assignment;
      found  : BOOLEAN;
      apoa   : BAG OF applied_organization_assignment;
      pdf    : SET OF product_definition_formation;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    pdf := product_definition_formation;
```



```
    REPEAT i := 1 TO SIZEOF(pdf) BY 1;
      apos := USEDIN(pdf[i],'TECHNICAL_DATA_PACKAGING.' +
          'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(apos) BY 1;
        IF EXISTS(apos[j].role) THEN
          IF (apos[j].role.name = 'creator') OR (apos[j].role.name =
              'id owner') THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF found THEN
        result := TRUE;
        ESCAPE;
      END_IF;
      apoa := USEDIN(pdf[i],'TECHNICAL_DATA_PACKAGING.' +
          'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS');
      REPEAT j := 1 TO SIZEOF(apoa) BY 1;
        IF EXISTS(apoa[j].role) THEN
          IF (apoa[j].role.name = 'creator') OR (apoa[j].role.name =
              'id owner') THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF NOT found THEN
        result := FALSE;
        ESCAPE;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- product_version_requires_person_organization

  RULE reference_document_requires_subcategorization FOR (
             product_related_product_category);

    LOCAL
      prpc   : SET OF product_related_product_category;
      pcr    : BAG OF product_category_relationship;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    prpc := QUERY ( v <* product_related_product_category | (v\
        product_category.name = 'reference document') );
    REPEAT i := 1 TO SIZEOF(prpc) BY 1;
      pcr := USEDIN(prpc[i],'TECHNICAL_DATA_PACKAGING.' +
          'PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY');
      IF SIZEOF(pcr) < 1 THEN
        result := FALSE;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- reference_document_requires_subcategorization
```

```
RULE release_authentication_string_restriction FOR (approval);

  LOCAL
    ar      : STRING := 'release authentication';
    result1 : BOOLEAN := TRUE;
    result2 : BOOLEAN := TRUE;
    found   : BOOLEAN;
    aaa     : BAG OF applied_approval_assignment;
    adt     : BAG OF approval_date_time;
    appr    : SET OF approval;
    apo     : BAG OF approval_person_organization;
  END_LOCAL;
  appr := approval;
  REPEAT i := 1 TO SIZEOF(appr) BY 1;
    aaa := USEDIN(appr[i],'TECHNICAL_DATA_PACKAGING.' +
        'APPROVAL_ASSIGNMENT.ASSIGNED_APPROVAL');
    found := FALSE;
    REPEAT j := 1 TO SIZEOF(aaa) BY 1;
      IF EXISTS(aaa[j]\approval_assignment.role) THEN
        IF aaa[j].role.name = ar THEN
          found := TRUE;
        END_IF;
      END_IF;
    END_REPEAT;
    IF found THEN
      apo := USEDIN(appr[i],'TECHNICAL_DATA_PACKAGING.' +
          'APPROVAL_PERSON_ORGANIZATION.AUTHORIZED_APPROVAL');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(apo) BY 1;
        IF EXISTS(apo[j].role) THEN
          IF apo[j].role.role = ar THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF NOT found THEN
        result1 := FALSE;
      END_IF;
      adt := USEDIN(appr[i],'TECHNICAL_DATA_PACKAGING.' +
          'APPROVAL_DATE_TIME.DATED_APPROVAL');
      found := FALSE;
      REPEAT j := 1 TO SIZEOF(adt) BY 1;
        IF EXISTS(adt[j].role) THEN
          IF adt[j].role.name = ar THEN
            found := TRUE;
          END_IF;
        END_IF;
      END_REPEAT;
      IF NOT found THEN
        result2 := FALSE;
      END_IF;
    END_IF;
  END_REPEAT;

WHERE
  wr1: result1;
  wr2: result2;

END_RULE; -- release_authentication_string_restriction
```



```
RULE security_classification_date_string_restriction FOR (
          applied_security_classification_assignment);

  LOCAL
    asca    : SET OF applied_security_classification_assignment;
    tresult : BOOLEAN;
    ada     : BAG OF applied_date_assignment;
    adta    : BAG OF applied_date_and_time_assignment;
    result  : BOOLEAN := TRUE;
  END_LOCAL;
  asca := applied_security_classification_assignment;
  REPEAT i := 1 TO SIZEOF(asca) BY 1;
    adta := USEDIN(asca[i],'TECHNICAL_DATA_PACKAGING.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS');
    tresult := FALSE;
    REPEAT j := 1 TO SIZEOF(adta) BY 1;
      REPEAT k := 1 TO SIZEOF(adta[j].items) BY 1;
        IF adta[j].items[k] = asca[i] THEN
          IF EXISTS(adta[j]\date_and_time_assignment.role) THEN
            IF adta[j]\date_and_time_assignment.role.name =
                'classification date' THEN
              tresult := TRUE;
            END_IF;
          END_IF;
          result := result AND tresult;
        END_IF;
      END_REPEAT;
    END_REPEAT;
    ada := USEDIN(asca[i],'TECHNICAL_DATA_PACKAGING.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS');
    REPEAT j := 1 TO SIZEOF(ada) BY 1;
      REPEAT k := 1 TO SIZEOF(ada[j].items) BY 1;
        IF ada[j].items[k] = asca[i] THEN
          IF EXISTS(ada[j]\date_assignment.role) THEN
            IF ada[j]\date_assignment.role.name = 'classification date'
                THEN
              tresult := TRUE;
            END_IF;
          END_IF;
          result := result AND tresult;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

WHERE
  wr1: result;

END_RULE; -- security_classification_date_string_restriction

RULE simple_list_of_elements_constraint FOR (applied_document_reference);

  LOCAL
    dri    : SET OF document_reference_item;
    adr    : SET OF applied_document_reference;
    result : BOOLEAN := TRUE;
  END_LOCAL;
  adr := applied_document_reference;
  REPEAT i := 1 TO SIZEOF(adr) BY 1;
```

```
      IF EXISTS(adr[i]\document_reference.role) THEN
        IF adr[i]\document_reference.role.name = 'list of elements' THEN
          dri := adr[i].items;
          IF SIZEOF(dri) < 1 THEN
            result := FALSE;
          END_IF;
          REPEAT j := 1 TO SIZEOF(dri) BY 1;
            IF NOT (
                'TECHNICAL_DATA_PACKAGING.PRODUCT_DEFINITION_FORMATION' IN
                TYPEOF(dri[j])) THEN
              result := FALSE;
            END_IF;
          END_REPEAT;
        END_IF;
      END_IF;
    END_REPEAT;

  WHERE
    wr1: result;

  END_RULE; -- simple_list_of_elements_constraint

  RULE simple_list_of_files_constraint FOR (applied_document_reference);

    LOCAL
      drt    : SET OF document_representation_type;
      dri    : SET OF document_reference_item;
      adr    : SET OF applied_document_reference;
      nameok : BOOLEAN;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    adr := applied_document_reference;
    REPEAT i := 1 TO SIZEOF(adr) BY 1;
      IF EXISTS(adr[i]\document_reference.role) THEN
        IF adr[i]\document_reference.role.name = 'list of files' THEN
          dri := adr[i].items;
          IF SIZEOF(dri) < 1 THEN
            result := FALSE;
          END_IF;
          REPEAT j := 1 TO SIZEOF(dri) BY 1;
            IF NOT ('TECHNICAL_DATA_PACKAGING.DOCUMENT_FILE' IN TYPEOF(dri
                [j])) THEN
              result := FALSE;
            END_IF;
            drt := dri[j].representation_types;
            nameok := FALSE;
            REPEAT k := 1 TO SIZEOF(drt) BY 1;
              IF (drt[k].name = 'digital') OR (drt[k].name = 'physical')
                   THEN
                nameok := TRUE;
              END_IF;
            END_REPEAT;
            IF NOT nameok THEN
              result := FALSE;
            END_IF;
          END_REPEAT;
        END_IF;
      END_IF;
    END_REPEAT;
```



```
WHERE
  wr1: result;

END_RULE; -- simple_list_of_files_constraint

RULE text_literal_alignment_baseline_constraint FOR (text_literal);

WHERE
  wr1: (SIZEOF(QUERY ( t <* text_literal | (NOT (t.alignment IN ['left',
          'center','right'])) )) = 0);

END_RULE; -- text_literal_alignment_baseline_constraint

RULE text_literal_font_requires_externally_defined_text_font FOR (
          text_literal);

WHERE
  wr1: (SIZEOF(QUERY ( t <* text_literal | (NOT (
          'TECHNICAL_DATA_PACKAGING.EXTERNALLY_DEFINED_TEXT_FONT' IN
          TYPEOF(t.font))) )) = 0);

END_RULE; -- text_literal_font_requires_externally_defined_text_font

RULE versioned_action_request_requires_status FOR (action_request_status,
          versioned_action_request);

WHERE
  wr1: (SIZEOF(QUERY ( ar <* versioned_action_request | (NOT (SIZEOF(
          QUERY ( ars <* action_request_status | (ar :=: ars.
          assigned_request) )) = 1)) )) = 0);

END_RULE; -- versioned_action_request_requires_status

FUNCTION acyclic_composite_text(
            start_composite: composite_text;
            child_text: SET [1:?] OF text_or_character
    ): LOGICAL;

  LOCAL
    i                    : INTEGER;
    local_annotation_text : SET [0:?] OF annotation_text;
    local_composite_text  : SET [0:?] OF composite_text;
    local_children        : SET [0:?] OF text_or_character;
  END_LOCAL;
  local_composite_text := QUERY ( child <* child_text | (
      'TECHNICAL_DATA_PACKAGING.COMPOSITE_TEXT' IN TYPEOF(child)) );
  IF SIZEOF(local_composite_text) > 0 THEN
    REPEAT i := 1 TO HIINDEX(local_composite_text) BY 1;
      IF start_composite :=: local_composite_text[i] THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  local_children := child_text;
  IF SIZEOF(local_composite_text) > 0 THEN
    REPEAT i := 1 TO HIINDEX(local_composite_text) BY 1;
      local_children := local_children + local_composite_text[i].
          collected_text;
    END_REPEAT;
```

```
    END_IF;
    local_annotation_text := QUERY ( child <* child_text | (
        'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT' IN TYPEOF(child)) );
    IF SIZEOF(local_annotation_text) > 0 THEN
      REPEAT i := 1 TO HIINDEX(local_annotation_text) BY 1;
        local_children := local_children + QUERY ( item <*
            local_annotation_text[i]\mapped_item.mapping_source.
            mapped_representation.items | (SIZEOF([
            'TECHNICAL_DATA_PACKAGING.ANNOTATION_TEXT',
            'TECHNICAL_DATA_PACKAGING.COMPOSITE_TEXT'] * TYPEOF(item)) > 0)
);
      END_REPEAT;
    END_IF;
    IF local_children :<>: child_text THEN
      RETURN(acyclic_composite_text(start_composite,local_children));
    ELSE
      RETURN(TRUE);
    END_IF;

  END_FUNCTION; -- acyclic_composite_text

  FUNCTION acyclic_curve_replica(
              rep: curve_replica;
              parent: curve
      ): BOOLEAN;
    IF NOT ('TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(parent))
         THEN
      RETURN(TRUE);
    END_IF;
    IF parent :=: rep THEN
      RETURN(FALSE);
    ELSE
      RETURN(acyclic_curve_replica(rep,parent\curve_replica.parent_curve));
    END_IF;

  END_FUNCTION; -- acyclic_curve_replica

  FUNCTION acyclic_mapped_representation(
              parent_set: SET OF representation;
              children_set: SET OF representation_item
      ): BOOLEAN;

    LOCAL
      x : SET OF representation_item;
      y : SET OF representation_item;
    END_LOCAL;
    x := QUERY ( z <* children_set | (
        'TECHNICAL_DATA_PACKAGING.MAPPED_ITEM' IN TYPEOF(z)) );
    IF SIZEOF(x) > 0 THEN
      REPEAT i := 1 TO HIINDEX(x) BY 1;
        IF x[i]\mapped_item.mapping_source.mapped_representation IN
            parent_set THEN
          RETURN(FALSE);
        END_IF;
        IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
            .mapping_source.mapped_representation,x[i]\mapped_item.
            mapping_source.mapped_representation.items) THEN
          RETURN(FALSE);
        END_IF;
```



```
    END_REPEAT;
  END_IF;
  x := children_set - x;
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      y := QUERY ( z <* bag_to_set(USEDIN(x[i],'')) | (
          'TECHNICAL_DATA_PACKAGING.REPRESENTATION_ITEM' IN TYPEOF(z)) );
      IF NOT acyclic_mapped_representation(parent_set,y) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- acyclic_mapped_representation

FUNCTION acyclic_point_replica(
            rep: point_replica;
            parent: point
    ): BOOLEAN;
  IF NOT ('TECHNICAL_DATA_PACKAGING.POINT_REPLICA' IN TYPEOF(parent))
      THEN
    RETURN(TRUE);
  END_IF;
  IF parent :=: rep THEN
    RETURN(FALSE);
  ELSE
    RETURN(acyclic_point_replica(rep,parent\point_replica.parent_pt));
  END_IF;

END_FUNCTION; -- acyclic_point_replica

FUNCTION acyclic_product_category_relationship(
            relation: product_category_relationship;
            children: SET OF product_category
    ): BOOLEAN;

  LOCAL
    x              : SET OF product_category_relationship;
    local_children : SET OF product_category;
  END_LOCAL;
  REPEAT i := 1 TO HIINDEX(children) BY 1;
    IF relation.category :=: children[i] THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  x := bag_to_set(USEDIN(relation.category,'TECHNICAL_DATA_PACKAGING.' +
      'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));
  local_children := children + relation.category;
  IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      IF NOT acyclic_product_category_relationship(x[i],local_children)
          THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(TRUE);
```

```
  END_FUNCTION; -- acyclic_product_category_relationship

  FUNCTION acyclic_product_definition_relationship(
              relation: product_definition_relationship;
              relatives: SET [1:?] OF product_definition;
              specific_relation: STRING
     ): BOOLEAN;

    LOCAL
      x : SET OF product_definition_relationship;
    END_LOCAL;
    IF relation.relating_product_definition IN relatives THEN
      RETURN(FALSE);
    END_IF;
    x := QUERY ( pd <* bag_to_set(USEDIN(relation.
        relating_product_definition,'TECHNICAL_DATA_PACKAGING.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION'))
         | (specific_relation IN TYPEOF(pd)) );
    REPEAT i := 1 TO HIINDEX(x) BY 1;
      IF NOT acyclic_product_definition_relationship(x[i],relatives +
          relation.relating_product_definition,specific_relation) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
    RETURN(TRUE);

  END_FUNCTION; -- acyclic_product_definition_relationship

  FUNCTION acyclic_solid_replica(
              rep: solid_replica;
              parent: solid_model
     ): BOOLEAN;
    IF NOT ('TECHNICAL_DATA_PACKAGING.SOLID_REPLICA' IN TYPEOF(parent))
        THEN
      RETURN(TRUE);
    END_IF;
    IF parent :=: rep THEN
      RETURN(FALSE);
    ELSE
      RETURN(acyclic_solid_replica(rep,parent\solid_replica.parent_solid));
    END_IF;

  END_FUNCTION; -- acyclic_solid_replica

  FUNCTION acyclic_surface_replica(
              rep: surface_replica;
              parent: surface
     ): BOOLEAN;
    IF NOT ('TECHNICAL_DATA_PACKAGING.SURFACE_REPLICA' IN TYPEOF(parent))
        THEN
      RETURN(TRUE);
    END_IF;
    IF parent :=: rep THEN
      RETURN(FALSE);
    ELSE
      RETURN(acyclic_surface_replica(rep,parent\surface_replica.
          parent_surface));
    END_IF;
```



```
END_FUNCTION; -- acyclic_surface_replica

FUNCTION aspect_ratio(
            p: planar_box
    ): positive_ratio_measure;
  IF (p.size_in_x > 0) AND (p.size_in_y > 0) THEN
    RETURN(p.size_in_x / p.size_in_y);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- aspect_ratio

FUNCTION associated_surface(
            arg: pcurve_or_surface
    ): surface;

  LOCAL
    surf : surface;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(arg) THEN
    surf := arg.basis_surface;
  ELSE
    surf := arg;
  END_IF;
  RETURN(surf);

END_FUNCTION; -- associated_surface

FUNCTION bag_to_set(
            the_bag: BAG OF GENERIC:intype
    ): SET OF GENERIC:intype;

  LOCAL
    the_set : SET OF GENERIC:intype := [];
  END_LOCAL;
  IF SIZEOF(the_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(the_bag) BY 1;
      the_set := the_set + the_bag[i];
    END_REPEAT;
  END_IF;
  RETURN(the_set);

END_FUNCTION; -- bag_to_set

FUNCTION base_axis(
            dim: INTEGER;
            axis1, axis2, axis3: direction
    ): LIST [2:3] OF direction;

  LOCAL
    u      : LIST [2:3] OF direction;
    d1     : direction;
    d2     : direction;
    factor : REAL;
  END_LOCAL;
  IF dim = 3 THEN
    d1 := NVL(normalise(axis3),dummy_gri || direction([0,0,1]));
    d2 := first_proj_axis(d1,axis1);
```

```
      u := [d2,second_proj_axis(d1,d2,axis2),d1];
    ELSE
      IF EXISTS(axis1) THEN
        d1 := normalise(axis1);
        u := [d1,orthogonal_complement(d1)];
        IF EXISTS(axis2) THEN
          factor := dot_product(axis2,u[2]);
          IF factor < 0 THEN
            u[2].direction_ratios[1] := -u[2].direction_ratios[1];
            u[2].direction_ratios[2] := -u[2].direction_ratios[2];
          END_IF;
        END_IF;
      ELSE
        IF EXISTS(axis2) THEN
          d1 := normalise(axis2);
          u := [orthogonal_complement(d1),d1];
          u[1].direction_ratios[1] := -u[1].direction_ratios[1];
          u[1].direction_ratios[2] := -u[1].direction_ratios[2];
        ELSE
          u := [dummy_gri || direction([1,0]),dummy_gri ||
direction([0,1])];
        END_IF;
      END_IF;
    END_IF;
    RETURN(u);

  END_FUNCTION; -- base_axis

  FUNCTION boolean_choose(
              b: BOOLEAN;
              choice1, choice2: GENERIC:item
    ): GENERIC:item;
    IF b THEN
      RETURN(choice1);
    ELSE
      RETURN(choice2);
    END_IF;

  END_FUNCTION; -- boolean_choose

  FUNCTION build_2axes(
              ref_direction: direction
    ): LIST [2:2] OF direction;

    LOCAL
      d : direction := NVL(normalise(ref_direction),dummy_gri ||
           direction([1,0]));
    END_LOCAL;
    RETURN([d,orthogonal_complement(d)]);

  END_FUNCTION; -- build_2axes

  FUNCTION build_axes(
              axis, ref_direction: direction
    ): LIST [3:3] OF direction;

    LOCAL
      d1 : direction;
      d2 : direction;
```



```
  END_LOCAL;
  d1 := NVL(normalise(axis),dummy_gri || direction([0,0,1]));
  d2 := first_proj_axis(d1,ref_direction);
  RETURN([d2,normalise(cross_product(d1,d2)).orientation,d1]);

END_FUNCTION; -- build_axes

FUNCTION closed_shell_reversed(
            a_shell: closed_shell
    ): oriented_closed_shell;

  LOCAL
    the_reverse : oriented_closed_shell;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.ORIENTED_CLOSED_SHELL' IN TYPEOF(a_shell)
       THEN
    the_reverse := dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces) || closed_shell() ||
        oriented_closed_shell(a_shell\oriented_closed_shell.
        closed_shell_element,NOT a_shell\oriented_closed_shell.
        orientation);
  ELSE
    the_reverse := dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces) || closed_shell() ||
        oriented_closed_shell(a_shell,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- closed_shell_reversed

FUNCTION conditional_reverse(
            p: BOOLEAN;
            an_item: reversible_topology
    ): reversible_topology;
  IF p THEN
    RETURN(an_item);
  ELSE
    RETURN(topology_reversed(an_item));
  END_IF;

END_FUNCTION; -- conditional_reverse

FUNCTION constraints_composite_curve_on_surface(
            c: composite_curve_on_surface
    ): BOOLEAN;

  LOCAL
    n_segments : INTEGER := SIZEOF(c.segments);
  END_LOCAL;
  REPEAT k := 1 TO n_segments BY 1;
    IF (NOT ('TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) AND (NOT (
        'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) AND (NOT (
        'TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c
        \composite_curve.segments[k].parent_curve))) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
```

```
    RETURN(TRUE);

  END_FUNCTION; -- constraints_composite_curve_on_surface

  FUNCTION constraints_geometry_shell_based_surface_model(
              m: shell_based_surface_model
        ): BOOLEAN;

    LOCAL
      result : BOOLEAN := TRUE;
    END_LOCAL;
    REPEAT j := 1 TO SIZEOF(m.sbsm_boundary) BY 1;
      IF (NOT ('TECHNICAL_DATA_PACKAGING.OPEN_SHELL' IN TYPEOF(m.
          sbsm_boundary[j]))) AND (NOT (
          'TECHNICAL_DATA_PACKAGING.CLOSED_SHELL' IN TYPEOF(m.sbsm_boundary
          [j]))) THEN
        result := FALSE;
        RETURN(result);
      END_IF;
    END_REPEAT;
    RETURN(result);

  END_FUNCTION; -- constraints_geometry_shell_based_surface_model

  FUNCTION constraints_geometry_shell_based_wireframe_model(
              m: shell_based_wireframe_model
         ): BOOLEAN;

    LOCAL
      result : BOOLEAN := TRUE;
    END_LOCAL;
    REPEAT j := 1 TO SIZEOF(m.sbwm_boundary) BY 1;
      IF (NOT ('TECHNICAL_DATA_PACKAGING.WIRE_SHELL' IN TYPEOF(m.
          sbwm_boundary[j]))) AND (NOT (
          'TECHNICAL_DATA_PACKAGING.VERTEX_SHELL' IN TYPEOF(m.sbwm_boundary
          [j]))) THEN
        result := FALSE;
        RETURN(result);
      END_IF;
    END_REPEAT;
    RETURN(result);

  END_FUNCTION; -- constraints_geometry_shell_based_wireframe_model

  FUNCTION constraints_param_b_spline(
              degree, up_knots, up_cp: INTEGER;
              knot_mult: LIST OF INTEGER;
              knots: LIST OF parameter_value
      ): BOOLEAN;

    LOCAL
      k      : INTEGER;
      sum    : INTEGER;
      result : BOOLEAN := TRUE;
    END_LOCAL;
    sum := knot_mult[1];
    REPEAT i := 2 TO up_knots BY 1;
      sum := sum + knot_mult[i];
    END_REPEAT;
```



```
  IF (degree < 1) OR (up_knots < 2) OR (up_cp < degree) OR (sum <> (
      degree + up_cp + 2)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  k := knot_mult[1];
  IF (k < 1) OR (k > (degree + 1)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  REPEAT i := 2 TO up_knots BY 1;
    IF (knot_mult[i] < 1) OR (knots[i] <= knots[i - 1]) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
    k := knot_mult[i];
    IF (i < up_knots) AND (k > degree) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
    IF (i = up_knots) AND (k > (degree + 1)) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- constraints_param_b_spline

FUNCTION constraints_rectangular_composite_surface(
            s: rectangular_composite_surface
    ): BOOLEAN;
  REPEAT i := 1 TO s.n_u BY 1;
    REPEAT j := 1 TO s.n_v BY 1;
      IF NOT (('TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE' IN TYPEOF(s.
          segments[i][j].parent_surface)) OR (
          'TECHNICAL_DATA_PACKAGING.RECTANGULAR_TRIMMED_SURFACE' IN
          TYPEOF(s.segments[i][j].parent_surface))) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_REPEAT;
  REPEAT i := 1 TO s.n_u - 1 BY 1;
    REPEAT j := 1 TO s.n_v BY 1;
      IF s.segments[i][j].u_transition = discontinuous THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_REPEAT;
  REPEAT i := 1 TO s.n_u BY 1;
    REPEAT j := 1 TO s.n_v - 1 BY 1;
      IF s.segments[i][j].v_transition = discontinuous THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION; -- constraints_rectangular_composite_surface
```

```
  FUNCTION cross_product(
              arg1, arg2: direction
      ): vector;

    LOCAL
      v2     : LIST [3:3] OF REAL;
      v1     : LIST [3:3] OF REAL;
      mag    : REAL;
      res    : direction;
      result : vector;
    END_LOCAL;
    IF (NOT EXISTS(arg1)) OR (arg1.dim = 2) OR (NOT EXISTS(arg2)) OR (arg2
        .dim = 2) THEN
      RETURN(?);
    ELSE
      BEGIN
        v1 := normalise(arg1).direction_ratios;
        v2 := normalise(arg2).direction_ratios;
        res := dummy_gri || direction([(v1[2] * v2[3]) - (v1[3] * v2[2]),(
            v1[3] * v2[1]) - (v1[1] * v2[3]),(v1[1] * v2[2]) - (v1[2] * v2[
            1])]);
        mag := 0;
        REPEAT i := 1 TO 3 BY 1;
          mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
        END_REPEAT;
        IF mag > 0 THEN
          result := dummy_gri || vector(res,SQRT(mag));
        ELSE
          result := dummy_gri || vector(arg1,0);
        END_IF;
        RETURN(result);
      END;
    END_IF;

  END_FUNCTION; -- cross_product

  FUNCTION curve_weights_positive(
              b: rational_b_spline_curve
      ): BOOLEAN;

    LOCAL
      result : BOOLEAN := TRUE;
    END_LOCAL;
    REPEAT i := 0 TO b.upper_index_on_control_points BY 1;
      IF b.weights[i] <= 0 THEN
        result := FALSE;
        RETURN(result);
      END_IF;
    END_REPEAT;
    RETURN(result);

  END_FUNCTION; -- curve_weights_positive

  FUNCTION derive_dimensional_exponents(
              x: unit
      ): dimensional_exponents;

    LOCAL
      result : dimensional_exponents := dimensional_exponents(0,0,0,0,0,0,
```



```
            0);
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.DERIVED_UNIT' IN TYPEOF(x) THEN
    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements) BY 1;
      result.length_exponent := result.length_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.length_exponent);
      result.mass_exponent := result.mass_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.mass_exponent);
      result.time_exponent := result.time_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.time_exponent);
      result.electric_current_exponent := result.
          electric_current_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.electric_current_exponent);
      result.thermodynamic_temperature_exponent := result.
          thermodynamic_temperature_exponent + (x.elements[i].exponent *
          x.elements[i].unit.dimensions.
          thermodynamic_temperature_exponent);
      result.amount_of_substance_exponent := result.
          amount_of_substance_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.amount_of_substance_exponent);
      result.luminous_intensity_exponent := result.
          luminous_intensity_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
  ELSE
    result := x.dimensions;
  END_IF;
  RETURN(result);

END_FUNCTION; -- derive_dimensional_exponents

FUNCTION dimension_of(
            item: geometric_representation_item
    ): dimension_count;

  LOCAL
    x   : SET OF representation;
    y   : representation_context;
    dim : dimension_count;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(item) THEN
    dim := SIZEOF(item\cartesian_point.coordinates);
    RETURN(dim);
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.DIRECTION' IN TYPEOF(item) THEN
    dim := SIZEOF(item\direction.direction_ratios);
    RETURN(dim);
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(item) THEN
    dim := SIZEOF(item\vector.orientation\direction.direction_ratios);
    RETURN(dim);
  END_IF;
  x := using_representations(item);
  y := x[1].context_of_items;
  dim := y\geometric_representation_context.coordinate_space_dimension;
  RETURN(dim);

END_FUNCTION; -- dimension_of
```

```
  FUNCTION dimensions_for_si_unit(
              n: si_unit_name
      ): dimensional_exponents;
    CASE n OF
      metre          :          RETURN(dimensional_exponents(1,0,0,0,0,0,0));
      gram           :          RETURN(dimensional_exponents(0,1,0,0,0,0,0));
      second         :          RETURN(dimensional_exponents(0,0,1,0,0,0,0));
      ampere         :          RETURN(dimensional_exponents(0,0,0,1,0,0,0));
      kelvin         :          RETURN(dimensional_exponents(0,0,0,0,1,0,0));
      mole           :          RETURN(dimensional_exponents(0,0,0,0,0,1,0));
      candela        :          RETURN(dimensional_exponents(0,0,0,0,0,0,1));
      radian         :          RETURN(dimensional_exponents(0,0,0,0,0,0,0));
      steradian      :          RETURN(dimensional_exponents(0,0,0,0,0,0,0));
      hertz          :          RETURN
                                   (dimensional_exponents(0,0,-1,0,0,0,0));
      newton         :          RETURN
                                   (dimensional_exponents(1,1,-2,0,0,0,0));
      pascal         :          RETURN
                                   (dimensional_exponents(-1,1,-2,0,0,0,0));
      joule          :          RETURN
                                   (dimensional_exponents(2,1,-2,0,0,0,0));
      watt           :          RETURN
                                   (dimensional_exponents(2,1,-3,0,0,0,0));
      coulomb        :          RETURN(dimensional_exponents(0,0,1,1,0,0,0));
      volt           :          RETURN
                                   (dimensional_exponents(2,1,-3,-1,0,0,0));
      farad          :          RETURN
                                   (dimensional_exponents(-2,-1,4,1,0,0,0));
      ohm            :          RETURN
                                   (dimensional_exponents(2,1,-3,-2,0,0,0));
      siemens        :          RETURN
                                   (dimensional_exponents(-2,-1,3,2,0,0,0));
      weber          :          RETURN
                                   (dimensional_exponents(2,1,-2,-1,0,0,0));
      tesla          :          RETURN
                                   (dimensional_exponents(0,1,-2,-1,0,0,0));
      henry          :          RETURN
                                   (dimensional_exponents(2,1,-2,-2,0,0,0));
      degree_celsius  :           RETURN
                                     (dimensional_exponents(0,0,0,0,1,0,0));
      lumen          :          RETURN(dimensional_exponents(0,0,0,0,0,0,1));
      lux            :          RETURN
                                    (dimensional_exponents(-2,0,0,0,0,0,1));
      becquerel      :          RETURN
                                    (dimensional_exponents(0,0,-1,0,0,0,0));
      gray           :          RETURN
                                    (dimensional_exponents(2,0,-2,0,0,0,0));
      sievert        :          RETURN
                                    (dimensional_exponents(2,0,-2,0,0,0,0));
      OTHERWISE      :          RETURN(?);
      END_CASE;

  END_FUNCTION; -- dimensions_for_si_unit

  FUNCTION dot_product(
              arg1, arg2: direction
      ): REAL;

    LOCAL
```



```
    ndim   : INTEGER;
    scalar : REAL;
    vec1   : direction;
    vec2   : direction;
  END_LOCAL;
  IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) THEN
    scalar := ?;
  ELSE
    IF arg1.dim <> arg2.dim THEN
      scalar := ?;
    ELSE
      BEGIN
        vec1 := normalise(arg1);
        vec2 := normalise(arg2);
        ndim := arg1.dim;
        scalar := 0;
        REPEAT i := 1 TO ndim BY 1;
          scalar := scalar + (vec1.direction_ratios[i] * vec2.
              direction_ratios[i]);
        END_REPEAT;
      END;
    END_IF;
  END_IF;
  RETURN(scalar);

END_FUNCTION; -- dot_product

FUNCTION edge_reversed(
            an_edge: edge
    ): oriented_edge;

  LOCAL
    the_reverse : oriented_edge;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.ORIENTED_EDGE' IN TYPEOF(an_edge) THEN
    the_reverse := dummy_tri || edge(an_edge.edge_end,an_edge.edge_start)
        || oriented_edge(an_edge\oriented_edge.edge_element,NOT an_edge\
        oriented_edge.orientation);
  ELSE
    the_reverse := dummy_tri || edge(an_edge.edge_end,an_edge.edge_start)
        || oriented_edge(an_edge,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- edge_reversed

FUNCTION face_bound_reversed(
            a_face_bound: face_bound
    ): face_bound;

  LOCAL
    the_reverse : face_bound;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.FACE_OUTER_BOUND' IN TYPEOF(a_face_bound)
       THEN
    the_reverse := dummy_tri || face_bound(a_face_bound\face_bound.bound,
        NOT a_face_bound\face_bound.orientation) || face_outer_bound();
  ELSE
    the_reverse := dummy_tri || face_bound(a_face_bound.bound,NOT
```



```
          a_face_bound.orientation);
    END_IF;
    RETURN(the_reverse);

  END_FUNCTION; -- face_bound_reversed

  FUNCTION face_reversed(
              a_face: face
      ): oriented_face;

    LOCAL
      the_reverse : oriented_face;
    END_LOCAL;
    IF 'TECHNICAL_DATA_PACKAGING.ORIENTED_FACE' IN TYPEOF(a_face) THEN
      the_reverse := dummy_tri || face(set_of_topology_reversed(a_face.
          bounds)) || oriented_face(a_face\oriented_face.face_element,NOT
          a_face\oriented_face.orientation);
    ELSE
      the_reverse := dummy_tri || face(set_of_topology_reversed(a_face.
          bounds)) || oriented_face(a_face,FALSE);
    END_IF;
    RETURN(the_reverse);

  END_FUNCTION; -- face_reversed

  FUNCTION first_proj_axis(
              z_axis, arg: direction
      ): direction;

    LOCAL
      x_vec  : vector;
      v      : direction;
      z      : direction;
      x_axis : direction;
    END_LOCAL;
    IF NOT EXISTS(z_axis) THEN
      RETURN(?);
    ELSE
      z := normalise(z_axis);
      IF NOT EXISTS(arg) THEN
        IF z.direction_ratios <> [1,0,0] THEN
          v := dummy_gri || direction([1,0,0]);
        ELSE
          v := dummy_gri || direction([0,1,0]);
        END_IF;
      ELSE
        IF arg.dim <> 3 THEN
          RETURN(?);
        END_IF;
        IF cross_product(arg,z).magnitude = 0 THEN
          RETURN(?);
        ELSE
          v := normalise(arg);
        END_IF;
      END_IF;
      x_vec := scalar_times_vector(dot_product(v,z),z);
      x_axis := vector_difference(v,x_vec).orientation;
      x_axis := normalise(x_axis);
    END_IF;
```



```
  RETURN(x_axis);

END_FUNCTION; -- first_proj_axis

FUNCTION gbsf_check_curve(
            cv: representation_item
    ): BOOLEAN;
  IF SIZEOF(['TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE',
      'TECHNICAL_DATA_PACKAGING.CONIC',
      'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA',
      'TECHNICAL_DATA_PACKAGING.LINE',
      'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1 THEN
    RETURN(FALSE);
  END_IF;
  IF SIZEOF(['TECHNICAL_DATA_PACKAGING.CIRCLE',
      'TECHNICAL_DATA_PACKAGING.ELLIPSE',
      'TECHNICAL_DATA_PACKAGING.TRIMMED_CURVE'] * TYPEOF(cv)) = 1 THEN
    RETURN(TRUE);
  ELSE
    IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (
        cv\b_spline_curve.self_intersect = FALSE)) OR (cv\b_spline_curve.
        self_intersect = UNKNOWN) THEN
      RETURN(TRUE);
    ELSE
      IF (('TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE' IN TYPEOF(cv)) AND
          (cv\composite_curve.self_intersect = FALSE)) OR (cv\
          composite_curve.self_intersect = UNKNOWN) THEN
        RETURN(SIZEOF(QUERY ( seg <* cv\composite_curve.segments | (NOT
            gbsf_check_curve(seg.parent_curve)) )) = 0);
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(cv) THEN
          RETURN(gbsf_check_curve(cv\curve_replica.parent_curve));
        ELSE
          IF ('TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D' IN TYPEOF(cv))
              AND ((cv\offset_curve_3d.self_intersect = FALSE) OR (cv\
              offset_curve_3d.self_intersect = UNKNOWN)) AND (NOT (
              'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv\
              offset_curve_3d.basis_curve))) THEN
            RETURN(gbsf_check_curve(cv\offset_curve_3d.basis_curve));
          ELSE
            IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(cv) THEN
              RETURN(gbsf_check_curve(cv\pcurve.reference_to_curve\
                  representation.items[1]) AND gbsf_check_surface(cv\
                  pcurve.basis_surface));
            ELSE
              IF 'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv) THEN
                IF SIZEOF(cv\polyline.points) >= 3 THEN
                  RETURN(TRUE);
                END_IF;
              ELSE
                IF 'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN TYPEOF(cv)
                     THEN
                  IF gbsf_check_curve(cv\surface_curve.curve_3d) THEN
                    REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                        associated_geometry) BY 1;
                      IF 'TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(cv
                          \surface_curve.associated_geometry[i]) THEN
                        IF NOT gbsf_check_surface(cv\surface_curve.
                            associated_geometry[i]) THEN
```

```
                          RETURN(FALSE);
                        END_IF;
                      ELSE
                        IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(
                            cv\surface_curve.associated_geometry[i])
                             THEN
                          IF NOT gbsf_check_curve(cv\surface_curve.
                              associated_geometry[i]) THEN
                            RETURN(FALSE);
                          END_IF;
                        END_IF;
                      END_IF;
                    END_REPEAT;
                    RETURN(TRUE);
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- gbsf_check_curve

  FUNCTION gbsf_check_point(
              pnt: point
      ): BOOLEAN;
    IF 'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
      RETURN(TRUE);
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE' IN TYPEOF(pnt) THEN
        RETURN(gbsf_check_curve(pnt\point_on_curve.basis_curve));
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.POINT_ON_SURFACE' IN TYPEOF(pnt)
             THEN
          RETURN(gbsf_check_surface(pnt\point_on_surface.basis_surface));
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.DEGENERATE_PCURVE' IN TYPEOF(pnt)
               THEN
            RETURN(gbsf_check_curve(pnt\degenerate_pcurve.
                reference_to_curve\representation.items[1]) AND
                gbsf_check_surface(pnt\degenerate_pcurve.basis_surface));
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- gbsf_check_point

  FUNCTION gbsf_check_surface(
              sf: surface
      ): BOOLEAN;
    IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE' IN TYPEOF(sf)) AND (
        sf\b_spline_surface.self_intersect = FALSE)) OR (sf\
        b_spline_surface.self_intersect = UNKNOWN) THEN
```



```
    RETURN(TRUE);
  ELSE
    IF SIZEOF(['TECHNICAL_DATA_PACKAGING.SPHERICAL_SURFACE',
        'TECHNICAL_DATA_PACKAGING.TOROIDAL_SURFACE',
        'TECHNICAL_DATA_PACKAGING.CURVE_BOUNDED_SURFACE',
        'TECHNICAL_DATA_PACKAGING.RECTANGULAR_TRIMMED_SURFACE'] * TYPEOF(
        sf)) = 1 THEN
      RETURN(TRUE);
    ELSE
      IF (('TECHNICAL_DATA_PACKAGING.OFFSET_SURFACE' IN TYPEOF(sf)) AND
          (sf\offset_surface.self_intersect = FALSE)) OR (sf\
          offset_surface.self_intersect = UNKNOWN) THEN
        RETURN(gbsf_check_surface(sf\offset_surface.basis_surface));
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.RECTANGULAR_COMPOSITE_SURFACE' IN
            TYPEOF(sf) THEN
          REPEAT i := 1 TO SIZEOF(sf\rectangular_composite_surface.
              segments) BY 1;
            REPEAT j := 1 TO SIZEOF(sf\rectangular_composite_surface.
                segments[i]) BY 1;
              IF NOT gbsf_check_surface(sf\rectangular_composite_surface
                  .segments[i][j].parent_surface) THEN
                RETURN(FALSE);
              END_IF;
            END_REPEAT;
          END_REPEAT;
          RETURN(TRUE);
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.SURFACE_REPLICA' IN TYPEOF(sf)
               THEN
            RETURN(gbsf_check_surface(sf\surface_replica
           .parent_surface));
          ELSE
            IF 'TECHNICAL_DATA_PACKAGING.SURFACE_OF_REVOLUTION' IN
                TYPEOF(sf) THEN
              RETURN(gbsf_check_curve(sf\swept_surface.swept_curve));
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- gbsf_check_surface

FUNCTION get_basis_surface(
             c: curve_on_surface
    ): SET [0:2] OF surface;

  LOCAL
    surfs : SET [0:2] OF surface;
    n     : INTEGER;
  END_LOCAL;
  surfs := [];
  IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN TYPEOF(c) THEN
```

```
      n := SIZEOF(c\surface_curve.associated_geometry);
      REPEAT i := 1 TO n BY 1;
        surfs := surfs + associated_surface(c\surface_curve.
            associated_geometry[i]);
      END_REPEAT;
    END_IF;
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c)
       THEN
    n := SIZEOF(c\composite_curve.segments);
    surfs := get_basis_surface(c\composite_curve.segments[1].
        parent_curve);
    IF n > 1 THEN
      REPEAT i := 2 TO n BY 1;
        surfs := surfs * get_basis_surface(c\composite_curve.segments[i]
            .parent_curve);
      END_REPEAT;
    END_IF;
  END_IF;
  RETURN(surfs);

END_FUNCTION; -- get_basis_surface

FUNCTION get_description_value(
             obj: description_attribute_select
    ): text;

  LOCAL
    description_bag : BAG OF description_attribute := USEDIN(obj,
                        'TECHNICAL_DATA_PACKAGING.' +
                        'DESCRIPTION_ATTRIBUTE.' + 'DESCRIBED_ITEM');
  END_LOCAL;
  IF SIZEOF(description_bag) = 1 THEN
    RETURN(description_bag[1].attribute_value);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- get_description_value

FUNCTION get_id_value(
             obj: id_attribute_select
    ): identifier;

  LOCAL
    id_bag : BAG OF id_attribute := USEDIN(obj,
              'TECHNICAL_DATA_PACKAGING.' + 'ID_ATTRIBUTE.' +
              'IDENTIFIED_ITEM');
  END_LOCAL;
  IF SIZEOF(id_bag) = 1 THEN
    RETURN(id_bag[1].attribute_value);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- get_id_value

FUNCTION get_name_value(
             obj: name_attribute_select
```



```
    ): label;

  LOCAL
    name_bag : BAG OF name_attribute := USEDIN(obj,
                'TECHNICAL_DATA_PACKAGING.' + 'NAME_ATTRIBUTE.' +
                'NAMED_ITEM');
  END_LOCAL;
  IF SIZEOF(name_bag) = 1 THEN
    RETURN(name_bag[1].attribute_value);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- get_name_value

FUNCTION get_role(
            obj: role_select
    ): object_role;

  LOCAL
    role_bag : BAG OF role_association := USEDIN(obj,
                'TECHNICAL_DATA_PACKAGING.' + 'ROLE_ASSOCIATION.' +
                'ITEM_WITH_ROLE');
  END_LOCAL;
  IF SIZEOF(role_bag) = 1 THEN
    RETURN(role_bag[1].role);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- get_role

FUNCTION item_in_context(
            item: representation_item;
            cntxt: representation_context
    ): BOOLEAN;

  LOCAL
    y : BAG OF representation_item;
  END_LOCAL;
  IF SIZEOF(USEDIN(item,'TECHNICAL_DATA_PACKAGING.REPRESENTATION.ITEMS')
      * cntxt.representations_in_context) > 0 THEN
    RETURN(TRUE);
  ELSE
    y := QUERY ( z <* USEDIN(item,'') | (
        'TECHNICAL_DATA_PACKAGING.REPRESENTATION_ITEM' IN TYPEOF(z)) );
    IF SIZEOF(y) > 0 THEN
      REPEAT i := 1 TO HIINDEX(y) BY 1;
        IF item_in_context(y[i],cntxt) THEN
          RETURN(TRUE);
        END_IF;
      END_REPEAT;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- item_in_context

FUNCTION leap_year(
```

```
              year: year_number
      ): BOOLEAN;
    IF (((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR ((year MOD 400) =
        0) THEN
      RETURN(TRUE);
    ELSE
      RETURN(FALSE);
    END_IF;

  END_FUNCTION; -- leap_year

  FUNCTION list_face_loops(
              f: face
      ): LIST [0:?] OF loop;

    LOCAL
      loops : LIST [0:?] OF loop := [];
    END_LOCAL;
    REPEAT i := 1 TO SIZEOF(f.bounds) BY 1;
      loops := loops + f.bounds[i].bound;
    END_REPEAT;
    RETURN(loops);

  END_FUNCTION; -- list_face_loops

  FUNCTION list_of_topology_reversed(
              a_list: list_of_reversible_topology_item
      ): list_of_reversible_topology_item;

    LOCAL
      the_reverse : list_of_reversible_topology_item;
    END_LOCAL;
    the_reverse := [];
    REPEAT i := 1 TO SIZEOF(a_list) BY 1;
      the_reverse := topology_reversed(a_list[i]) + the_reverse;
    END_REPEAT;
    RETURN(the_reverse);

  END_FUNCTION; -- list_of_topology_reversed

  FUNCTION list_to_array(
              lis: LIST [0:?] OF GENERIC:t;
              low, u: INTEGER
      ): ARRAY OF GENERIC:t;
    LOCAL
      n   : INTEGER;
      res : ARRAY [low:u] OF GENERIC:t;
    END_LOCAL;
    n := SIZEOF(lis);
    IF n <> ((u - low) + 1) THEN
      RETURN(?);
    ELSE
      res := [lis[1],n];
      REPEAT i := 2 TO n BY 1;
        res[(low + i) - 1] := lis[i];
      END_REPEAT;
      RETURN(res);
    END_IF;
  END_FUNCTION; -- list_to_array
```



```
FUNCTION list_to_set(
            l: LIST [0:?] OF GENERIC:t
    ): SET OF GENERIC:t;

  LOCAL
    s : SET OF GENERIC:t := [];
  END_LOCAL;
  REPEAT i := 1 TO SIZEOF(l) BY 1;
    s := s + l[i];
  END_REPEAT;
  RETURN(s);

END_FUNCTION; -- list_to_set

FUNCTION make_array_of_array(
            lis: LIST [1:?] OF LIST [1:?] OF GENERIC:t;
            low1, u1, low2, u2: INTEGER
    ): ARRAY OF ARRAY OF GENERIC:t;

  LOCAL
    res : ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;
  END_LOCAL;
  IF ((u1 - low1) + 1) <> SIZEOF(lis) THEN
    RETURN(?);
  END_IF;
  IF ((u2 - low2) + 1) <> SIZEOF(lis[1]) THEN
    RETURN(?);
  END_IF;
  res := [list_to_array(lis[1],low2,u2),(u1 - low1) + 1];
  REPEAT i := 2 TO HIINDEX(lis) BY 1;
    IF ((u2 - low2) + 1) <> SIZEOF(lis[i]) THEN
      RETURN(?);
    END_IF;
    res[(low1 + i) - 1] := list_to_array(lis[i],low2,u2);
  END_REPEAT;
  RETURN(res);

END_FUNCTION; -- make_array_of_array

FUNCTION mixed_loop_type_set(
            l: SET [0:?] OF loop
    ): LOGICAL;

  LOCAL
    poly_loop_type : LOGICAL;
  END_LOCAL;
  IF SIZEOF(l) <= 1 THEN
    RETURN(FALSE);
  END_IF;
  poly_loop_type := 'TECHNICAL_DATA_PACKAGING.POLY_LOOP' IN TYPEOF(l[1]);
  REPEAT i := 2 TO SIZEOF(l) BY 1;
    IF ('TECHNICAL_DATA_PACKAGING.POLY_LOOP' IN TYPEOF(l[i])) <>
        poly_loop_type THEN
      RETURN(TRUE);
    END_IF;
  END_REPEAT;
  RETURN(FALSE);
```

```
  END_FUNCTION; -- mixed_loop_type_set

  FUNCTION msb_shells(
              brep: manifold_solid_brep
     ): SET [1:?] OF closed_shell;
    IF SIZEOF(QUERY ( msbtype <* TYPEOF(brep) | (msbtype LIKE
       '*BREP_WITH_VOIDS') )) >= 1 THEN
      RETURN(brep\brep_with_voids.voids + brep.outer);
    ELSE
      RETURN([brep.outer]);
    END_IF;

  END_FUNCTION; -- msb_shells

  FUNCTION msf_curve_check(
              cv: representation_item
     ): BOOLEAN;
    IF SIZEOF(['TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE',
       'TECHNICAL_DATA_PACKAGING.CONIC',
       'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA',
       'TECHNICAL_DATA_PACKAGING.LINE',
       'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1 THEN
      RETURN(FALSE);
    END_IF;
    IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv\
       b_spline_curve.self_intersect = FALSE)) OR (cv\b_spline_curve.
       self_intersect = UNKNOWN) THEN
      RETURN(TRUE);
    ELSE
      IF SIZEOF(['TECHNICAL_DATA_PACKAGING.CONIC',
          'TECHNICAL_DATA_PACKAGING.LINE'] * TYPEOF(cv)) = 1 THEN
        RETURN(TRUE);
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(cv) THEN
          RETURN(msf_curve_check(cv\curve_replica.parent_curve));
        ELSE
          IF ('TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D' IN TYPEOF(cv))
              AND ((cv\offset_curve_3d.self_intersect = FALSE) OR (cv\
              offset_curve_3d.self_intersect = UNKNOWN)) AND (NOT (
              'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv\
              offset_curve_3d.basis_curve))) THEN
            RETURN(msf_curve_check(cv\offset_curve_3d.basis_curve));
          ELSE
            IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(cv) THEN
              RETURN(msf_curve_check(cv\pcurve.reference_to_curve\
                  representation.items[1]) AND msf_surface_check(cv\pcurve.
                  basis_surface));
            ELSE
              IF 'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN TYPEOF(cv)
                   THEN
                IF msf_curve_check(cv\surface_curve.curve_3d) THEN
                  REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                      associated_geometry) BY 1;
                    IF 'TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(cv\
                        surface_curve.associated_geometry[i]) THEN
                      IF NOT msf_surface_check(cv\surface_curve.
                          associated_geometry[i]) THEN
                        RETURN(FALSE);
                      END_IF;
```



```
                    ELSE
                      IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(cv\
                          surface_curve.associated_geometry[i]) THEN
                        IF NOT msf_curve_check(cv\surface_curve.
                            associated_geometry[i]) THEN
                          RETURN(FALSE);
                        END_IF;
                      END_IF;
                    END_IF;
                  END_REPEAT;
                  RETURN(TRUE);
                END_IF;
              ELSE
                IF 'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv) THEN
                  IF SIZEOF(cv\polyline.points) >= 3 THEN
                    RETURN(TRUE);
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- msf_curve_check

  FUNCTION msf_surface_check(
              surf: surface
      ): BOOLEAN;
    IF 'TECHNICAL_DATA_PACKAGING.ELEMENTARY_SURFACE' IN TYPEOF(surf) THEN
      RETURN(TRUE);
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.SWEPT_SURFACE' IN TYPEOF(surf) THEN
        RETURN(msf_curve_check(surf\swept_surface.swept_curve));
      ELSE
        IF (('TECHNICAL_DATA_PACKAGING.OFFSET_SURFACE' IN TYPEOF(surf))
            AND (surf\offset_surface.self_intersect = FALSE)) OR (surf\
            offset_surface.self_intersect = UNKNOWN) THEN
          RETURN(msf_surface_check(surf\offset_surface.basis_surface));
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.SURFACE_REPLICA' IN TYPEOF(surf)
              THEN
            RETURN(msf_surface_check(surf\surface_replica.parent_surface));
          ELSE
            IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE' IN TYPEOF(
                surf)) AND (surf\b_spline_surface.self_intersect = FALSE))
                OR (surf\b_spline_surface.self_intersect = UNKNOWN) THEN
              RETURN(TRUE);
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- msf_surface_check
```

```
  FUNCTION nmsf_curve_check(
              cv: representation_item
      ): BOOLEAN;
    IF SIZEOF(['TECHNICAL_DATA_PACKAGING.BOUNDED_CURVE',
        'TECHNICAL_DATA_PACKAGING.CONIC',
        'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA',
        'TECHNICAL_DATA_PACKAGING.LINE',
        'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1 THEN
      RETURN(FALSE);
    ELSE
      IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (
          cv\b_spline_curve.self_intersect = FALSE)) OR (cv\b_spline_curve.
          self_intersect = UNKNOWN) THEN
        RETURN(TRUE);
      ELSE
        IF SIZEOF(['TECHNICAL_DATA_PACKAGING.CONIC',
            'TECHNICAL_DATA_PACKAGING.LINE'] * TYPEOF(cv)) = 1 THEN
          RETURN(TRUE);
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(cv) THEN
            RETURN(nmsf_curve_check(cv\curve_replica.parent_curve));
          ELSE
            IF ('TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D' IN TYPEOF(cv))
                AND ((cv\offset_curve_3d.self_intersect = FALSE) OR (cv\
                offset_curve_3d.self_intersect = UNKNOWN)) AND (NOT (
                'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv\
                offset_curve_3d.basis_curve))) THEN
              RETURN(nmsf_curve_check(cv\offset_curve_3d.basis_curve));
            ELSE
              IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(cv) THEN
                RETURN(nmsf_curve_check(cv\pcurve.reference_to_curve\
                    representation.items[1]) AND nmsf_surface_check(cv\
                    pcurve.basis_surface));
              ELSE
                IF 'TECHNICAL_DATA_PACKAGING.SURFACE_CURVE' IN TYPEOF(cv)
                     THEN
                  IF nmsf_curve_check(cv\surface_curve.curve_3d) THEN
                    REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                        associated_geometry) BY 1;
                      IF 'TECHNICAL_DATA_PACKAGING.SURFACE' IN TYPEOF(cv\
                          surface_curve.associated_geometry[i]) THEN
                        IF NOT nmsf_surface_check(cv\surface_curve.
                            associated_geometry[i]) THEN
                          RETURN(FALSE);
                        END_IF;
                      ELSE
                        IF 'TECHNICAL_DATA_PACKAGING.PCURVE' IN TYPEOF(cv\
                            surface_curve.associated_geometry[i]) THEN
                          IF NOT nmsf_curve_check(cv\surface_curve.
                              associated_geometry[i]) THEN
                            RETURN(FALSE);
                          END_IF;
                        END_IF;
                      END_IF;
                    END_REPEAT;
                    RETURN(TRUE);
                  END_IF;
                ELSE
                  IF 'TECHNICAL_DATA_PACKAGING.POLYLINE' IN TYPEOF(cv)
```



```
                    THEN
                  IF SIZEOF(cv\polyline.points) >= 3 THEN
                    RETURN(TRUE);
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- nmsf_curve_check

FUNCTION nmsf_surface_check(
             surf: surface
    ): BOOLEAN;
  IF 'TECHNICAL_DATA_PACKAGING.ELEMENTARY_SURFACE' IN TYPEOF(surf) THEN
    RETURN(TRUE);
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.SWEPT_SURFACE' IN TYPEOF(surf) THEN
      RETURN(nmsf_curve_check(surf\swept_surface.swept_curve));
    ELSE
      IF (('TECHNICAL_DATA_PACKAGING.OFFSET_SURFACE' IN TYPEOF(surf))
          AND (surf\offset_surface.self_intersect = FALSE)) OR (surf\
          offset_surface.self_intersect = UNKNOWN) THEN
        RETURN(nmsf_surface_check(surf\offset_surface.basis_surface));
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.SURFACE_REPLICA' IN TYPEOF(surf)
            THEN
          RETURN(nmsf_surface_check(surf\surface_replica
          .parent_surface));
        ELSE
          IF (('TECHNICAL_DATA_PACKAGING.B_SPLINE_SURFACE' IN TYPEOF(
              surf)) AND (surf\b_spline_surface.self_intersect = FALSE))
              OR (surf\b_spline_surface.self_intersect = UNKNOWN) THEN
            RETURN(TRUE);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- nmsf_surface_check

FUNCTION normalise(
             arg: vector_or_direction
    ): vector_or_direction;

  LOCAL
    ndim   : INTEGER;
    v      : direction;
    vec    : vector;
    mag    : REAL;
    result : vector_or_direction;
  END_LOCAL;
```

```
    IF NOT EXISTS(arg) THEN
      result := ?;
    ELSE
      ndim := arg.dim;
      IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(arg) THEN
        BEGIN
          v := dummy_gri || direction(arg.orientation.direction_ratios);
          IF arg.magnitude = 0 THEN
            RETURN(?);
          ELSE
            vec := dummy_gri || vector(v,1);
          END_IF;
        END;
      ELSE
        v := dummy_gri || direction(arg.direction_ratios);
      END_IF;
      mag := 0;
      REPEAT i := 1 TO ndim BY 1;
        mag := mag + (v.direction_ratios[i] * v.direction_ratios[i]);
      END_REPEAT;
      IF mag > 0 THEN
        mag := SQRT(mag);
        REPEAT i := 1 TO ndim BY 1;
          v.direction_ratios[i] := v.direction_ratios[i] / mag;
        END_REPEAT;
        IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(arg) THEN
          vec.orientation := v;
          result := vec;
        ELSE
          result := v;
        END_IF;
      ELSE
        RETURN(?);
      END_IF;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- normalise

  FUNCTION open_shell_reversed(
              a_shell: open_shell
      ): oriented_open_shell;

    LOCAL
      the_reverse : oriented_open_shell;
    END_LOCAL;
    IF 'TECHNICAL_DATA_PACKAGING.ORIENTED_OPEN_SHELL' IN TYPEOF(a_shell)
         THEN
      the_reverse := dummy_tri || connected_face_set(a_shell\
          connected_face_set.cfs_faces) || open_shell() ||
          oriented_open_shell(a_shell\oriented_open_shell.
          open_shell_element,NOT a_shell\oriented_open_shell.orientation);
    ELSE
      the_reverse := dummy_tri || connected_face_set(a_shell\
          connected_face_set.cfs_faces) || open_shell() ||
          oriented_open_shell(a_shell,FALSE);
    END_IF;
    RETURN(the_reverse);
  END_FUNCTION; -- open_shell_reversed
```



```
FUNCTION orthogonal_complement(
            vec: direction
    ): direction;

  LOCAL
    result : direction;
  END_LOCAL;
  IF (vec.dim <> 2) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
  ELSE
    result := dummy_gri || direction([-vec.direction_ratios[2],vec.
        direction_ratios[1]]);
    RETURN(result);
  END_IF;

END_FUNCTION; -- orthogonal_complement

FUNCTION path_head_to_tail(
            a_path: path
    ): BOOLEAN;

  LOCAL
    n : INTEGER;
    p : BOOLEAN := TRUE;
  END_LOCAL;
  n := SIZEOF(a_path.edge_list);
  REPEAT i := 2 TO n BY 1;
    p := p AND (a_path.edge_list[i - 1].edge_end :=: a_path.edge_list[i]
        .edge_start);
  END_REPEAT;
  RETURN(p);

END_FUNCTION; -- path_head_to_tail

FUNCTION path_reversed(
            a_path: path
    ): oriented_path;

  LOCAL
    the_reverse : oriented_path;
  END_LOCAL;
  IF 'TECHNICAL_DATA_PACKAGING.ORIENTED_PATH' IN TYPEOF(a_path) THEN
    the_reverse := dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list)) || oriented_path(a_path\oriented_path.path_element,
        NOT a_path\oriented_path.orientation);
  ELSE
    the_reverse := dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list)) || oriented_path(a_path,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- path_reversed

FUNCTION product_category_graph_members(
            p: product_category
    ): SET OF product_category;
```

```
    LOCAL
      pcr    : BAG OF product_category_relationship;
      result : SET OF product_category := [];
    END_LOCAL;
    pcr := USEDIN(p,'TECHNICAL_DATA_PACKAGING.' +
        'PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY');
    IF SIZEOF(pcr) > 0 THEN
      result := result + p;
      REPEAT i := 1 TO SIZEOF(pcr) BY 1;
        result := result + product_category_graph_members(pcr[i].
            sub_category);
      END_REPEAT;
    END_IF;
    pcr := USEDIN(p,'TECHNICAL_DATA_PACKAGING.' +
        'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY');
    IF SIZEOF(pcr) > 0 THEN
      result := result + p;
      REPEAT i := 1 TO SIZEOF(pcr) BY 1;
        result := result + product_category_graph_members(pcr[i].category);
      END_REPEAT;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- product_category_graph_members

  FUNCTION scalar_times_vector(
              scalar: REAL;
              vec: vector_or_direction
      ): vector;

    LOCAL
      v      : direction;
      mag    : REAL;
      result : vector;
    END_LOCAL;
    IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
      RETURN(?);
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(vec) THEN
        v := dummy_gri || direction(vec.orientation.direction_ratios);
        mag := scalar * vec.magnitude;
      ELSE
        v := dummy_gri || direction(vec.direction_ratios);
        mag := scalar;
      END_IF;
      IF mag < 0 THEN
        REPEAT i := 1 TO SIZEOF(v.direction_ratios) BY 1;
          v.direction_ratios[i] := -v.direction_ratios[i];
        END_REPEAT;
        mag := -mag;
      END_IF;
      result := dummy_gri || vector(normalise(v),mag);
    END_IF;
    RETURN(result);

  END_FUNCTION; -- scalar_times_vector

  FUNCTION second_proj_axis(
              z_axis, x_axis, arg: direction
```



```
    ): direction;

  LOCAL
    temp   : vector;
    v      : direction;
    y_axis : vector;
  END_LOCAL;
  IF NOT EXISTS(arg) THEN
    v := dummy_gri || direction([0,1,0]);
  ELSE
    v := arg;
  END_IF;
  temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
  y_axis := vector_difference(v,temp);
  temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
  y_axis := vector_difference(y_axis,temp);
  y_axis := normalise(y_axis);
  RETURN(y_axis.orientation);

END_FUNCTION; -- second_proj_axis

FUNCTION set_of_topology_reversed(
            a_set: set_of_reversible_topology_item
    ): set_of_reversible_topology_item;

  LOCAL
    the_reverse : set_of_reversible_topology_item;
  END_LOCAL;
  the_reverse := [];
  REPEAT i := 1 TO SIZEOF(a_set) BY 1;
    the_reverse := the_reverse + topology_reversed(a_set[i]);
  END_REPEAT;
  RETURN(the_reverse);

END_FUNCTION; -- set_of_topology_reversed

FUNCTION shell_reversed(
            a_shell: shell
    ): shell;
  IF 'TECHNICAL_DATA_PACKAGING.OPEN_SHELL' IN TYPEOF(a_shell) THEN
    RETURN(open_shell_reversed(a_shell));
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.CLOSED_SHELL' IN TYPEOF(a_shell) THEN
      RETURN(closed_shell_reversed(a_shell));
    ELSE
      RETURN(?);
    END_IF;
  END_IF;

END_FUNCTION; -- shell_reversed

FUNCTION surface_weights_positive(
            b: rational_b_spline_surface
    ): BOOLEAN;

  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT i := 0 TO b.u_upper BY 1;
```

```
      REPEAT j := 0 TO b.v_upper BY 1;
        IF b.weights[i][j] <= 0 THEN
          result := FALSE;
          RETURN(result);
        END_IF;
      END_REPEAT;
    END_REPEAT;
    RETURN(result);

  END_FUNCTION; -- surface_weights_positive

  FUNCTION topology_reversed(
              an_item: reversible_topology
      ): reversible_topology;
    IF 'TECHNICAL_DATA_PACKAGING.EDGE' IN TYPEOF(an_item) THEN
      RETURN(edge_reversed(an_item));
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.PATH' IN TYPEOF(an_item) THEN
      RETURN(path_reversed(an_item));
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.FACE_BOUND' IN TYPEOF(an_item) THEN
      RETURN(face_bound_reversed(an_item));
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.FACE' IN TYPEOF(an_item) THEN
      RETURN(face_reversed(an_item));
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.SHELL' IN TYPEOF(an_item) THEN
      RETURN(shell_reversed(an_item));
    END_IF;
    IF 'SET' IN TYPEOF(an_item) THEN
      RETURN(set_of_topology_reversed(an_item));
    END_IF;
    IF 'LIST' IN TYPEOF(an_item) THEN
      RETURN(list_of_topology_reversed(an_item));
    END_IF;
    RETURN(?);

  END_FUNCTION; -- topology_reversed

  FUNCTION using_items(
              item: founded_item_select;
              checked_items: SET OF founded_item_select
      ): SET OF founded_item_select;

    LOCAL
      next_items      : SET OF founded_item_select;
      new_check_items : SET OF founded_item_select;
      result_items    : SET OF founded_item_select;
    END_LOCAL;
    result_items := [];
    new_check_items := checked_items + item;
    next_items := QUERY ( z <* bag_to_set(USEDIN(item,'')) | ((
        'TECHNICAL_DATA_PACKAGING.REPRESENTATION_ITEM' IN TYPEOF(z)) OR (
        'TECHNICAL_DATA_PACKAGING.FOUNDED_ITEM' IN TYPEOF(z))) );
    IF SIZEOF(next_items) > 0 THEN
      REPEAT i := 1 TO HIINDEX(next_items) BY 1;
        IF NOT (next_items[i] IN new_check_items) THEN
          result_items := result_items + next_items[i] + using_items(
              next_items[i],new_check_items);
```



```
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(result_items);

END_FUNCTION; -- using_items

FUNCTION using_representations(
            item: founded_item_select
    ): SET OF representation;

  LOCAL
    results            : SET OF representation;
    intermediate_items : SET OF founded_item_select;
    result_bag         : BAG OF representation;
  END_LOCAL;
  results := [];
  result_bag := USEDIN(item,
      'TECHNICAL_DATA_PACKAGING.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag) BY 1;
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  intermediate_items := using_items(item,[]);
  IF SIZEOF(intermediate_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(intermediate_items) BY 1;
      result_bag := USEDIN(intermediate_items[i],
          'TECHNICAL_DATA_PACKAGING.REPRESENTATION.ITEMS');
      IF SIZEOF(result_bag) > 0 THEN
        REPEAT j := 1 TO HIINDEX(result_bag) BY 1;
          results := results + result_bag[j];
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(results);

END_FUNCTION; -- using_representations

FUNCTION valid_basis_curve_in_2d_wireframe(
            crv: curve
    ): BOOLEAN;
  IF SIZEOF(['TECHNICAL_DATA_PACKAGING.POLYLINE',
      'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
      'TECHNICAL_DATA_PACKAGING.ELLIPSE',
      'TECHNICAL_DATA_PACKAGING.CIRCLE'] * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.TRIMMED_CURVE' IN TYPEOF(crv) THEN
      IF SIZEOF(['TECHNICAL_DATA_PACKAGING.LINE',
          'TECHNICAL_DATA_PACKAGING.PARABOLA',
          'TECHNICAL_DATA_PACKAGING.HYPERBOLA'] * TYPEOF(crv\
          trimmed_curve.basis_curve)) = 1 THEN
        RETURN(TRUE);
      ELSE
        RETURN(valid_basis_curve_in_2d_wireframe(crv\trimmed_curve.
            basis_curve));
      END_IF;
```

```
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_2D' IN TYPEOF(crv) THEN
          RETURN(valid_basis_curve_in_2d_wireframe(crv\offset_curve_2d.
              basis_curve));
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(crv) THEN
            RETURN(valid_basis_curve_in_2d_wireframe(crv\curve_replica.
                parent_curve));
          ELSE
            IF 'TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE' IN TYPEOF(crv)
                THEN
              RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments |
                  (NOT valid_basis_curve_in_2d_wireframe(ccs.parent_curve))
))
                  = 0);
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- valid_basis_curve_in_2d_wireframe

  FUNCTION valid_calendar_date(
              date: calendar_date
      ): LOGICAL;
    CASE date.month_component OF
      1 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 31));
      2 :         BEGIN
          IF leap_year(date.year_component) THEN
            RETURN((1 <= date.day_component) AND (date.day_component <=
29));
          ELSE
            RETURN((1 <= date.day_component) AND (date.day_component <=
28));
          END_IF;
        END;
      3 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 31));
      4 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 30));
      5 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 31));
      6 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 30));
      7 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 31));
      8 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 31));
      9 :         RETURN((1 <= date.day_component) AND (date.day_component
          <= 30));
      10  :         RETURN((1 <= date.day_component) AND (date.
        day_component <= 31));
      11  :         RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
      12  :         RETURN((1 <= date.day_component) AND (date.
        day_component <= 31));
```



```
    END_CASE;
  RETURN(FALSE);

END_FUNCTION; -- valid_calendar_date

FUNCTION valid_geometrically_bounded_wf_curve(
            crv: curve
    ): BOOLEAN;
  IF SIZEOF(['TECHNICAL_DATA_PACKAGING.POLYLINE',
      'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
      'TECHNICAL_DATA_PACKAGING.ELLIPSE',
      'TECHNICAL_DATA_PACKAGING.CIRCLE'] * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.TRIMMED_CURVE' IN TYPEOF(crv) THEN
      IF SIZEOF(['TECHNICAL_DATA_PACKAGING.LINE',
          'TECHNICAL_DATA_PACKAGING.PARABOLA',
          'TECHNICAL_DATA_PACKAGING.HYPERBOLA'] * TYPEOF(crv\
          trimmed_curve.basis_curve)) = 1 THEN
        RETURN(TRUE);
      ELSE
        RETURN(valid_geometrically_bounded_wf_curve(crv\trimmed_curve.
            basis_curve));
      END_IF;
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D' IN TYPEOF(crv) THEN
        RETURN(valid_geometrically_bounded_wf_curve(crv\offset_curve_3d.
            basis_curve));
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(crv) THEN
          RETURN(valid_geometrically_bounded_wf_curve(crv\curve_replica.
              parent_curve));
        ELSE
          IF 'TECHNICAL_DATA_PACKAGING.COMPOSITE_CURVE' IN TYPEOF(crv)
               THEN
            RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments |
                (NOT valid_geometrically_bounded_wf_curve(ccs.
                parent_curve)) )) = 0);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_curve

FUNCTION valid_geometrically_bounded_wf_point(
            pnt: point
    ): BOOLEAN;
  IF 'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
    RETURN(TRUE);
  ELSE
    IF 'TECHNICAL_DATA_PACKAGING.POINT_ON_CURVE' IN TYPEOF(pnt) THEN
      RETURN(valid_geometrically_bounded_wf_curve(pnt\point_on_curve.
          basis_curve));
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.POINT_REPLICA' IN TYPEOF(pnt) THEN
        RETURN(valid_geometrically_bounded_wf_point(pnt\point_replica.
```

```
            parent_pt));
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_point

FUNCTION valid_measure_value(
            m: measure_value
    ): BOOLEAN;
  IF 'REAL' IN TYPEOF(m) THEN
    RETURN(m > 0);
  ELSE
    IF 'INTEGER' IN TYPEOF(m) THEN
      RETURN(m > 0);
    ELSE
      RETURN(TRUE);
    END_IF;
  END_IF;

END_FUNCTION; -- valid_measure_value

FUNCTION valid_time(
            time: local_time
    ): BOOLEAN;
  IF EXISTS(time.second_component) THEN
    RETURN(EXISTS(time.minute_component));
  ELSE
    RETURN(TRUE);
  END_IF;

END_FUNCTION; -- valid_time

FUNCTION valid_units(
            m: measure_with_unit
    ): BOOLEAN;
  IF 'TECHNICAL_DATA_PACKAGING.LENGTH_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.MASS_MEASURE' IN TYPEOF(m.value_component)
       THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,1,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.TIME_MEASURE' IN TYPEOF(m.value_component)
       THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,1,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'TECHNICAL_DATA_PACKAGING.ELECTRIC_CURRENT_MEASURE' IN TYPEOF(m.
```



```
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,1,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.THERMODYNAMIC_TEMPERATURE_MEASURE' IN
    TYPEOF(m.value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,1,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.CELSIUS_TEMPERATURE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,1,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.AMOUNT_OF_SUBSTANCE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,1,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.LUMINOUS_INTENSITY_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,1) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.PLANE_ANGLE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.SOLID_ANGLE_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(0,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.AREA_MEASURE' IN TYPEOF(m.value_component)
     THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(2,0,0,0,0,0,0) THEN
    RETURN(FALSE);
  END_IF;
END_IF;
IF 'TECHNICAL_DATA_PACKAGING.VOLUME_MEASURE' IN TYPEOF(m.
    value_component) THEN
  IF derive_dimensional_exponents(m.unit_component) <>
      dimensional_exponents(3,0,0,0,0,0,0) THEN
```

```
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.RATIO_MEASURE' IN TYPEOF(m.
        value_component) THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.POSITIVE_LENGTH_MEASURE' IN TYPEOF(m.
        value_component) THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(1,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    IF 'TECHNICAL_DATA_PACKAGING.POSITIVE_PLANE_ANGLE_MEASURE' IN TYPEOF(m
        .value_component) THEN
      IF derive_dimensional_exponents(m.unit_component) <>
          dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
      END_IF;
    END_IF;
    RETURN(TRUE);

  END_FUNCTION; -- valid_units

  FUNCTION valid_wireframe_edge_curve(
              crv: curve
      ): BOOLEAN;
    IF SIZEOF(['TECHNICAL_DATA_PACKAGING.LINE',
        'TECHNICAL_DATA_PACKAGING.CONIC',
        'TECHNICAL_DATA_PACKAGING.B_SPLINE_CURVE',
        'TECHNICAL_DATA_PACKAGING.POLYLINE'] * TYPEOF(crv)) = 1 THEN
      RETURN(TRUE);
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.CURVE_REPLICA' IN TYPEOF(crv) THEN
        RETURN(valid_wireframe_edge_curve(crv\curve_replica.parent_curve));
      ELSE
        IF 'TECHNICAL_DATA_PACKAGING.OFFSET_CURVE_3D' IN TYPEOF(crv) THEN
          RETURN(valid_wireframe_edge_curve(crv\offset_curve_3d.
              basis_curve));
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- valid_wireframe_edge_curve

  FUNCTION valid_wireframe_vertex_point(
              pnt: point
      ): BOOLEAN;
    IF 'TECHNICAL_DATA_PACKAGING.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
      RETURN(TRUE);
    ELSE
      IF 'TECHNICAL_DATA_PACKAGING.POINT_REPLICA' IN TYPEOF(pnt) THEN
        RETURN(valid_wireframe_vertex_point(pnt\point_replica.parent_pt));
      END_IF;
```



```
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- valid_wireframe_vertex_point

  FUNCTION vector_difference(
              arg1, arg2: vector_or_direction
    ): vector;

    LOCAL
      ndim   : INTEGER;
      mag2   : REAL;
      mag1   : REAL;
      mag    : REAL;
      res    : direction;
      vec1   : direction;
      vec2   : direction;
      result : vector;
    END_LOCAL;
    IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) OR (arg1.dim <> arg2.dim)
        THEN
      RETURN(?);
    ELSE
      BEGIN
        IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(arg1) THEN
          mag1 := arg1.magnitude;
          vec1 := arg1.orientation;
        ELSE
          mag1 := 1;
          vec1 := arg1;
        END_IF;
        IF 'TECHNICAL_DATA_PACKAGING.VECTOR' IN TYPEOF(arg2) THEN
          mag2 := arg2.magnitude;
          vec2 := arg2.orientation;
        ELSE
          mag2 := 1;
          vec2 := arg2;
        END_IF;
        vec1 := normalise(vec1);
        vec2 := normalise(vec2);
        ndim := SIZEOF(vec1.direction_ratios);
        mag := 0;
        res := dummy_gri || direction(vec1.direction_ratios);
        REPEAT i := 1 TO ndim BY 1;
          res.direction_ratios[i] := (mag1 * vec1.direction_ratios[i]) + (
              mag2 * vec2.direction_ratios[i]);
          mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
        END_REPEAT;
        IF mag > 0 THEN
          result := dummy_gri || vector(res,SQRT(mag));
        ELSE
          result := dummy_gri || vector(vec1,0);
        END_IF;
      END;
    END_IF;
    RETURN(result);
  END_FUNCTION; -- vector_difference

END_SCHEMA; -- technical_data_packaging
```

Annex B
(normative)

AIM short names of entities

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 — AIM short names of entites

Entity name	Short name
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS	AMWAD
ACTION_PROPERTY	ACTPRP
ACTION_PROPERTY_REPRESENTATION	ACPRRP
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_RESOURCE	ACTRSR
ACTION_RESOURCE_RELATIONSHIP	ACRSRL
ACTION_RESOURCE_TYPE	ACRSTY
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ALTERNATE_PRODUCT_RELATIONSHIP	ALPRRL
AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT	AOSMWU
AMOUNT_OF_SUBSTANCE_UNIT	AOSU
ANNOTATION_CURVE_OCCURRENCE	ANCROC
ANNOTATION_OCCURRENCE	ANNOCC
ANNOTATION_POINT_OCCURRENCE	ANPNOC
ANNOTATION_TEXT	ANNTXT
ANNOTATION_TEXT_CHARACTER	ANTXCH
ANNOTATION_TEXT_OCCURRENCE	ANTXOC
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_CONTEXT_RELATIONSHIP	APCNRL
APPLICATION_PROTOCOL_DEFINITION	APPRDF

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
APPLIED_ACTION_ASSIGNMENT	APACAS
APPLIED_ACTION_REQUEST_ASSIGNMENT	AARA
APPLIED_APPROVAL_ASSIGNMENT	APAPAS
APPLIED_CERTIFICATION_ASSIGNMENT	APCRAS
APPLIED_CLASSIFICATION_ASSIGNMENT	APCLAS
APPLIED_CONTRACT_ASSIGNMENT	APCNAS
APPLIED_DATE_AND_TIME_ASSIGNMENT	ADATA
APPLIED_DATE_ASSIGNMENT	APDTAS
APPLIED_DOCUMENT_REFERENCE	APDCRF
APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	ADUCA
APPLIED_EFFECTIVITY_ASSIGNMENT	APEFAS
APPLIED_EFFECTIVITY_CONTEXT_ASSIGNMENT	AECA
APPLIED_EVENT_OCCURRENCE_ASSIGNMENT	AEOA
APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT	AEIA
APPLIED_IDENTIFICATION_ASSIGNMENT	APIDAS
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
APPLIED_ORGANIZATIONAL_PROJECT_ASSIGNMENT	AOPA
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA
APPLIED_PRESENTED_ITEM	APPRIT
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ASCA
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
APPROVAL_DATE_TIME	APDTTM
APPROVAL_PERSON_ORGANIZATION	APPROR
APPROVAL_RELATIONSHIP	APPRLT
APPROVAL_ROLE	APPRL
APPROVAL_STATUS	APPSTT
AREA_DEPENDENT_ANNOTATION_REPRESENTATION	ADAR
AREA_IN_SET	ARINST
AREA_MEASURE_WITH_UNIT	AMWU
AREA_UNIT	ARUNT
ASSEMBLY_COMPONENT_USAGE	ASCMUS
ASSEMBLY_COMPONENT_USAGE_SUBSTITUTE	ACUS
ASSEMBLY_COMPONENT_USAGE_SUBSTITUTE_WITH_RANKING	ACUSWR
AXIS1_PLACEMENT	AX1PLC
AXIS2_PLACEMENT_2D	A2PL2D
AXIS2_PLACEMENT_3D	A2PL3D
B_SPLINE_CURVE	BSPCR

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
B_SPLINE_CURVE_WITH_KNOTS	BSCWK
B_SPLINE_SURFACE	BSPSR
B_SPLINE_SURFACE_WITH_KNOTS	BSSWK
BACKGROUND_COLOUR	BCKCLR
BEZIER_CURVE	BZRCRV
BEZIER_SURFACE	BZRSRF
BLOCK	BLOCK
BOOLEAN_RESULT	BLNRSL
BOUNDARY_CURVE	BNDCR
BOUNDED_CURVE	BNDCRV
BOUNDED_PCURVE	BNDPCR
BOUNDED_SURFACE	BNDSRF
BOUNDED_SURFACE_CURVE	BNSRCR
BOX_DOMAIN	BXDMN
BOXED_HALF_SPACE	BXHLSP
BREP_WITH_VOIDS	BRWTVD
CALENDAR_DATE	CLNDT
CAMERA_IMAGE	CMRIMG
CAMERA_IMAGE_3D_WITH_SCALE	CI3WS
CAMERA_MODEL	CMRMDL
CAMERA_MODEL_D3	CMMDD3
CAMERA_MODEL_D3_WITH_HLHSR	CMDWH
CAMERA_MODEL_WITH_LIGHT_SOURCES	CMWLS
CAMERA_USAGE	CMRUSG
CARTESIAN_POINT	CRTPNT
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
CARTESIAN_TRANSFORMATION_OPERATOR_2D	CTO2
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3
CERTIFICATION	CRTFCT
CERTIFICATION_ASSIGNMENT	CRTASS
CERTIFICATION_TYPE	CRTTYP
CHARACTER_GLYPH_STYLE_OUTLINE	CGO
CHARACTER_GLYPH_STYLE_STROKE	CGS
CHARACTER_GLYPH_SYMBOL	CHGLSY
CHARACTERIZED_OBJECT	CHROBJ
CIRCLE	CIRCLE
CLASS	CLASS
CLASS_SYSTEM	CLS1

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
CLASSIFICATION_ASSIGNMENT	CLSASS
CLASSIFICATION_ROLE	CLSRL
CLOSED_SHELL	CLSSHL
COLOUR	COLOUR
COLOUR_RGB	CLRRGB
COLOUR_SPECIFICATION	CLRSPC
COMPOSITE_CURVE	CMPCRV
COMPOSITE_CURVE_ON_SURFACE	CCOS
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_TEXT	COMTXT
COMPOUND_REPRESENTATION_ITEM	CMRPIT
CONFIGURATION_DESIGN	CNFDSG
CONFIGURATION_EFFECTIVITY	CNFEFF
CONFIGURATION_ITEM	CNFITM
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_EDGE_SET	CNEDST
CONNECTED_FACE_SET	CNFCST
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	CDSR
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONTRACT	CNTRCT
CONTRACT_ASSIGNMENT	CNTASS
CONTRACT_RELATIONSHIP	CNTRLT
CONTRACT_TYPE	CNTTYP
CONVERSION_BASED_UNIT	CNBSUN
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
CSG_SHAPE_REPRESENTATION	CSSHRP
CSG_SOLID	CSGSLD
CURVE	CURVE
CURVE_BOUNDED_SURFACE	CRBNSR
CURVE_REPLICA	CRVRPL
CURVE_STYLE	CRVSTY
CURVE_STYLE_FONT	CRSTFN
CURVE_STYLE_FONT_PATTERN	CSFP
CURVE_STYLE_RENDERING	CRSTRN
CYLINDRICAL_SURFACE	CYLSRF
DATE	DATE
DATE_AND_TIME	DTANTM

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
DATE_AND_TIME_ASSIGNMENT	DATA
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATE_TIME_ROLE	DTTMRL
DATED_EFFECTIVITY	DTDEFF
DEFINED_CHARACTER_GLYPH	DFCHGL
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_PCURVE	DGNPCR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTION_ ATTRIBUTE	DSCATT
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
DESIGN_MAKE_FROM_RELATIONSHIP	DMFR
DIMENSIONAL_EXPONENTS	DMNEXP
DIRECTED_ACTION	DRCACT
DIRECTION	DRCTN
DOCUMENT	DCMNT
DOCUMENT_FILE	DCMFL
DOCUMENT_PRODUCT_ASSOCIATION	DCP1
DOCUMENT_PRODUCT_EQUIVALENCE	DCPREQ
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_REPRESENTATION_TYPE	DCRPTY
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	DUCA
DOCUMENT_USAGE_ROLE	DCUSRL
DRAUGHTING_APPROVAL_ASSIGNMENT	DRAPAS
DRAUGHTING_CALLOUT	DRGCLL
DRAUGHTING_DRAWING_REVISION	DRDRRV
DRAUGHTING_PRE_DEFINED_COLOUR	DPDC
DRAUGHTING_PRE_DEFINED_CURVE_FONT	DPDCF
DRAUGHTING_TITLE	DRGTTL
DRAWING_DEFINITION	DRWDFN
DRAWING_REVISION	DRWRVS
DRAWING_SHEET_REVISION	DRSHRV
EDGE	EDGE

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
EDGE_BASED_WIREFRAME_MODEL	EBWM
EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION	EBWSR
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EFFECTIVITY	EFFCTV
EFFECTIVITY_ASSIGNMENT	EFFASS
EFFECTIVITY_CONTEXT_ASSIGNMENT	EFC0
EFFECTIVITY_CONTEXT_ROLE	EFCNRL
EFFECTIVITY_RELATIONSHIP	EFFRLT
ELECTRIC_CURRENT_MEASURE_WITH_UNIT	ECMWU
ELECTRIC_CURRENT_UNIT	ELCRUN
ELEMENTARY_BREP_SHAPE_REPRESENTATION	EBSR
ELEMENTARY_SURFACE	ELMSRF
ELLIPSE	ELLPS
EVALUATED_DEGENERATE_PCURVE	EVDGPC
EVENT_OCCURRENCE	EVNOCC
EVENT_OCCURRENCE_ASSIGNMENT	EVOCAS
EVENT_OCCURRENCE_RELATIONSHIP	EVOO
EVENT_OCCURRENCE_ROLE	EVOCRL
EXECUTED_ACTION	EXCACT
EXTERNAL_IDENTIFICATION_ASSIGNMENT	EXIDAS
EXTERNAL_SOURCE	EXTSRC
EXTERNALLY_DEFINED_CHARACTER_GLYPH	EDCG
EXTERNALLY_DEFINED_CLASS	EXDO
EXTERNALLY_DEFINED_GENERAL_PROPERTY	EDGP
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_PLANAR_BOX	EDPB
EXTERNALLY_DEFINED_STYLE	EXDFST
EXTERNALLY_DEFINED_SYMBOL	EXDFSY
EXTERNALLY_DEFINED_SYMBOL_AND_PLACEMENT	EDSAP
EXTERNALLY_DEFINED_TEXT_FONT	EDTF
EXTRUDED_FACE_SOLID	EXFCSL
FACE	FACE
FACE_BASED_SURFACE_MODEL	FBSM
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SURFACE	FCSRF
FACETED_BREP	FCTBR

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
FACETED_BREP_SHAPE_REPRESENTATION	FBSR
FILL_AREA_STYLE	FLARST
FILL_AREA_STYLE_COLOUR	FASC
FOUNDED_ITEM	FNDITM
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
GENERAL_PROPERTY	GNRPRP
GENERAL_PROPERTY_ASSOCIATION	GNPRAS
GENERAL_PROPERTY_RELATIONSHIP	GNPRRL
GEOMETRIC_CURVE_SET	GMCRST
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_SET	GMTST
GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION	GB2WR
GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION	GBSSR
GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION	GBWSR
GLOBAL_UNCERTAINTLY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUP_RELATIONSHIP	GRPRLT
HALF_SPACE_SOLID	HLSPSL
HYPERBOLA	HYPRBL
ID_ATTRIBUTE	IDATT
IDENTIFICATION_ASSIGNMENT	IDNASS
IDENTIFICATION_ASSIGNMENT_RELATIONSHIP	IDASRL
IDENTIFICATION_ROLE	IDNRL
INTERSECTION_CURVE	INTCRV
INVISIBILITY	INVSBL
ITEM_DEFINED_TRANSFORMATION	ITDFTR
ITEM_IDENTIFIED_REPRESENTATION_USAGE	IIRU
LANGUAGE_ASSIGNMENT	LNGASS
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIBRARY_CONTEXT	LBRCNT
LIGHT_SOURCE	LGHSRC
LIGHT_SOURCE_AMBIENT	LGSRAM
LIGHT_SOURCE_DIRECTIONAL	LGSRDR
LIGHT_SOURCE_POSITIONAL	LGSRPS
LIGHT_SOURCE_SPOT	LGSRSP

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
LINE	LINE
LOCAL_TIME	LCLTM
LOOP	LOOP
LOT_EFFECTIVITY	LTEFF
LUMINOUS_INTENSITY_MEASURE_WITH_UNIT	LIMWU
LUMINOUS_INTENSITY_UNIT	LMINUN
MAKE_FROM_USAGE_OPTION	MFUO
MAKE_FROM_USAGE_OPTION_WITH_REFERENCE_DESIGNATOR	MFUOWR
MANIFOLD_SOLID_BREP	MNSLBR
MANIFOLD_SURFACE_SHAPE_REPRESENTATION	MSSR
MAPPED_ITEM	MPPITM
MASS_MEASURE_WITH_UNIT	MMWU
MASS_UNIT	MSSUNT
MATERIAL_DESIGNATION	MTRDSG
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_AREA	MDGPA
MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION	MDGPR
MECHANICAL_DESIGN_SHADED_PRESENTATION_AREA	MDSPA
MECHANICAL_DESIGN_SHADED_PRESENTATION_REPRESENTATION	MDSPR
NAME_ATTRIBUTE	NMATT
NAMED_UNIT	NMDUNT
NEXT_ASSEMBLY_USAGE_OCCURRENCE	NAUO
NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION	NMSSR
OBJECT_ROLE	OBJRL
OFFSET_CURVE_2D	OFCR2D
OFFSET_CURVE_3D	OFCR3D
OFFSET_SURFACE	OFFSRF
OPEN_SHELL	OPNSHL
ORGANIZATION	ORGNZT
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_RELATIONSHIP	ORGRLT
ORGANIZATION_ROLE	ORGRL
ORGANIZATIONAL_ADDRESS	ORGADD
ORGANIZATIONAL_PROJECT	ORGPRJ
ORGANIZATIONAL_PROJECT_ASSIGNMENT	ORPRAS
ORGANIZATIONAL_PROJECT_RELATIONSHIP	ORP0

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
ORGANIZATIONAL_PROJECT_ROLE	ORPRRL
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
OTHER_LIST_TABLE_REPRESENTATION	OLTR
OUTER_BOUNDARY_CURVE	OTBNCR
OVER_RIDING_STYLED_ITEM	ORSI
PARABOLA	PRBL
PARAMETRIC_REPRESENTATION_CONTEXT	PRRPCN
PATH	PATH
PCURVE	PCURVE
PERSON	PERSON
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSONAL_ADDRESS	PRSADD
PLACEMENT	PLCMNT
PLANAR_BOX	PLNBX
PLANAR_EXTENT	PLNEXT
PLANE	PLANE
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN
POINT	POINT
POINT_ON_CURVE	PNONCR
POINT_ON_SURFACE	PNONSR
POINT_REPLICA	PNTRPL
POINT_STYLE	PNTSTY
POLY_LOOP	PLYLP
POLYLINE	PLYLN
PRE_DEFINED_CHARACTER_GLYPH	PDCG
PRE_DEFINED_COLOUR	PRDFCL
PRE_DEFINED_CURVE_FONT	PDCF
PRE_DEFINED_ITEM	PRDFIT
PRE_DEFINED_TEXT_FONT	PDTF
PRECISION_QUALIFIER	PRCQLF
PRESENTATION_AREA	PRSAR

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
PRESENTATION_REPRESENTATION	PRSRPR
PRESENTATION_SET	PRSST
PRESENTATION_SIZE	PRSSZ
PRESENTATION_STYLE_ASSIGNMENT	PRSTAS
PRESENTATION_STYLE_BY_CONTEXT	PSBC
PRESENTATION_VIEW	PRSVW
PRESENTED_ITEM	PRSITM
PRESENTED_ITEM_REPRESENTATION	PRITRP
PROCESS_PRODUCT_ASSOCIATION	PRPRAS
PRODUCT	PRDCT
PRODUCT_CATEGORY	PRDCTG
PRODUCT_CATEGORY_RELATIONSHIP	PRCTRL
PRODUCT_CONCEPT	PRDCNC
PRODUCT_CONCEPT_CONTEXT	PRCNCN
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_CONTEXT_ASSOCIATION	PDCA
PRODUCT_DEFINITION_CONTEXT_ROLE	PDCR
PRODUCT_DEFINITION_EFFECTIVITY	PRDFEF
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE	PDFWSS
PRODUCT_DEFINITION_OCCURRENCE_RELATIONSHIP	PDOR
PRODUCT_DEFINITION_PROCESS	PRDFPR
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_SUBSTITUTE	PRDFSB
PRODUCT_DEFINITION_USAGE	PRDFUS
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PRODUCT_RELATIONSHIP	PRDRLT
PROMISSORY_USAGE_OCCURRENCE	PRUSOC
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
QUANTIFIED_ASSEMBLY_COMPONENT_USAGE	QACU

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS
RECTANGULAR_COMPOSITE_SURFACE	RCCMSR
RECTANGULAR_TRIMMED_SURFACE	RCTRSR
RELATIVE_EVENT_OCCURRENCE	RLEVOC
REPARAMETRISED_COMPOSITE_CURVE_SEGMENT	RCCS
REPRESENTATION	RPRSNT
REPRESENTATION_CONTEXT	RPRCNT
REPRESENTATION_ITEM	RPRITM
REPRESENTATION_ITEM_RELATIONSHIP	RPITRL
REPRESENTATION_MAP	RPRMP
REPRESENTATION_RELATIONSHIP	RPRRLT
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION	RRWT
REVOLVED_FACE_SOLID	RVFCSL
RIGHT_ANGULAR_WEDGE	RGANWD
RIGHT_CIRCULAR_CONE	RGCRCN
RIGHT_CIRCULAR_CYLINDER	RGCRCY
ROLE_ASSOCIATION	RLASS
SEAM_CURVE	SMCRV
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SERIAL_NUMBERED_EFFECTIVITY	SRNMEF
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_RELATIONSHIP	SHRJPRL
SHELL_BASED_SURFACE_MODEL	SBSM
SHELL_BASED_WIREFRAME_MODEL	SBWM
SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION	SBWSR
SI_UNIT	SUNT
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
SOLID_ANGLE_UNIT	SLANUN

Table B.1 — AIM short names of entities (continued)

Entity name	Short name
SOLID_MODEL	SLDMDL
SOLID_REPLICA	SLDRPL
SPECIFIED_HIGHER_USAGE_OCCURRENCE	SHUO
SPHERE	SPHERE
SPHERICAL_SURFACE	SPHSRF
STYLED_ITEM	STYITM
SURFACE	SRFC
SURFACE_CURVE	SRFCRV
SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PATCH	SRFPTC
SURFACE_RENDERING_PROPERTIES	SRRNPR
SURFACE_REPLICA	SRFRPL
SURFACE_SIDE_STYLE	SRSDST
SURFACE_STYLE_BOUNDARY	SRSTBN
SURFACE_STYLE_CONTROL_GRID	SSCG
SURFACE_STYLE_FILL_AREA	SSFA
SURFACE_STYLE_PARAMETER_LINE	SSPL
SURFACE_STYLE_REFLECTANCE_AMBIENT	SSRA
SURFACE_STYLE_REFLECTANCE_AMBIENT_DIFFUSE	SSRAD
SURFACE_STYLE_REFLECTANCE_AMBIENT_DIFFUSE_SPECULAR	SSRADS
SURFACE_STYLE_RENDERING	SRSTRN
SURFACE_STYLE_RENDERING_WITH_PROPERTIES	SSRWP
SURFACE_STYLE_SEGMENTATION_CURVE	SSSC
SURFACE_STYLE_SILHOUETTE	SRSTSL
SURFACE_STYLE_TRANSPARENT	SRSTTR
SURFACE_STYLE_USAGE	SRSTUS
SWEPT_FACE_SOLID	SWFCSL
SWEPT_SURFACE	SWPSRF
SYMBOL_REPRESENTATION	SYMRPR
TEXT_LITERAL	TXTLTR
TEXT_STRING_REPRESENTATION	TXSRP
TEXT_STYLE	TXTSTY
TEXT_STYLE_FOR_DEFINED_FONT	TSFDF
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	TTMWU
THERMODYNAMIC_TEMPERATURE_UNIT	THTMUN
TIME_INTERVAL	TMINT
TIME_INTERVAL_BASED_EFFECTIVITY	TIBE

Table B.1 - AIM short names of entities (concluded)

Entity name	Short name
TIME_INTERVAL_RELATIONSHIP	TMINRL
TIME_INTERVAL_WITH_BONDS	TIWB
TIME_MEASURE_WITH_UNIT	TMWU
TIME_UNIT	TMUNT
TOPOLOGICAL_REPRESENTATION_ITEM	TPRPIT
TOROIDAL_SURFACE	TRDSRF
TORUS	TORUS
TRIMMED_CURVE	TRMCRV
TYPE_QUALIFIER	TYPQLF
UNCERTAINTY_MEASURE_WITH_UNIT	UMWU
UNIFORM_CURVE	UNFCRV
UNIFORM_SURFACE	UNFSRF
VLAUE_REPRESENTATION_ITEM	VLRPIT
VECTOR	VECTOR
VERSIONED_ACTION_REQUEST	VRACRQ
VERTEX	VERTEX
VERTEX_LOOP	VRTLP
VERTEX_POINT	VRTPNT
VERTEX_SHELL	VRTSHL
VIEW_VOLUME	VWVLM
VOLUME_MEASURE_WITH_UNIT	VMWU
VOLUME_UNIT	VLMUNT
WIRE_SHELL	WRSHL

Annex C
(normative)

Implementation method specific requirements

C.1 Exchange Structure

The implementation method defines what types of exchange behavior are required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and in the AIM defined in annex A of this part of ISO 10303. The header exchange structure shall identify use of this part of ISO 10303 by the schema name technical_data_packaging.

C.2 Exchange structure for drawings and sheets with Identification Conformance Classes

The capability this part of ISO 10303 will provide is to allow a drawing and its sheets to be identified just like other Tdp_elements are so they can be managed and interpreted as configuration controlled objects in product data management systems. ISO 10303-201 and ISO 10303-202 represent drawings and sheets as presentations. This part of ISO 10303 will contain a portion of the same constructs as in ISO 10303-201 and ISO 10303-202 so that the identification of the drawing (draughting_drawing_revision) and sheets (drawing_sheet_revision) in ISO 10303-201 and ISO 10303-202 can be associated to the configuration controlled identification of a drawing and sheet (product_definition_formation) in this part of ISO 10303. Figure C.2 and Figure C.3 show these common constructs that are shared by this part of ISO 10303, ISO 10303-201, and ISO 10303-202. This capability could be implemented with each part of ISO 10303 (application protocol) contained in its own physical file or contained in a multi-schema physical file. In separate physical files the constructs in Figures C.2 and C.3 would be captured in each file. In one physical file the constructs in Figures C.2 and C.3 would only have to reside once. The following sub clauses identify different implementation examples.

C.2.1 Exchange structure for this part of ISO 10303

When this part of ISO 10303 is implemented independently of ISO 10303-201 or ISO 10303-202, the AIM constructs in Figure C.1 shall be used in the implementation. Using this implementation method for this part of ISO 10303 does not utilize ISO 10303-505 constructs.

The left half of Figure C.1 is for the relationship between two documents, **product**. The **product_definition_formation_relationship.relating_document** is the drawing. The **product_relationship.related_document** is the sheet of the drawing. The **product_relationship.name** is ‘drawing membership’.

The right half of Figure C.1 is for the relationship between two document_versions, **product_definition_formation**s. The **product_definition_formation_relationship.relating_document** is the drawing. The **product_definition_formation_relationship.related_document** is the sheet of the drawing. The **product_definition_formation_relationship.name** is ‘drawing membership’.

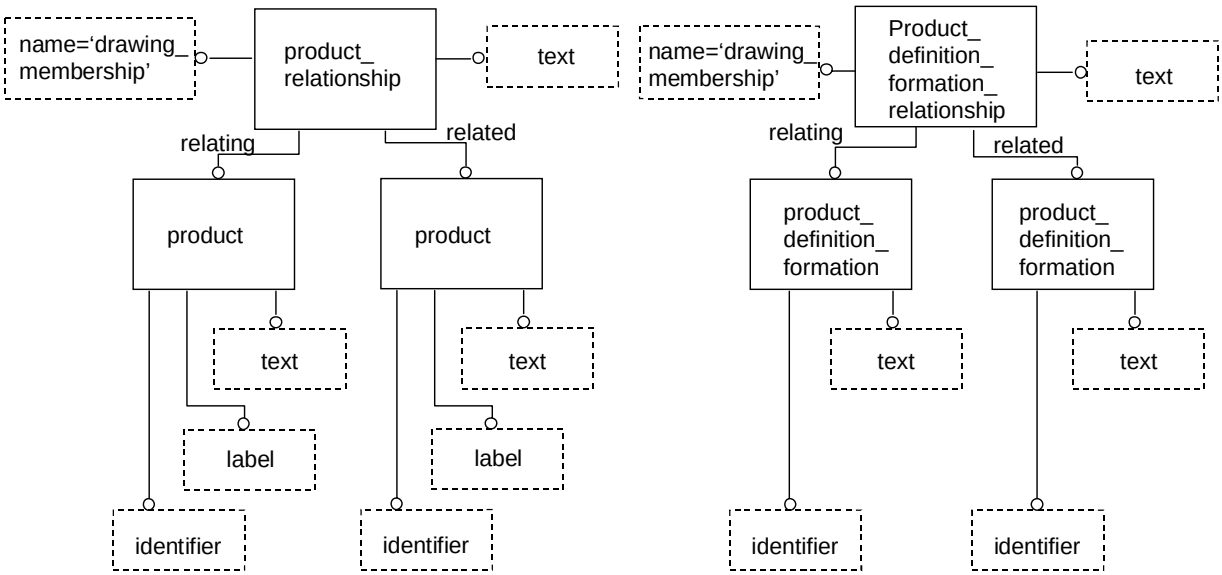

Figure C.1 — Exchange structure without ISO 10303-201 or ISO 10303-202

NOTE The graphical representation in Figure C.1 is presented in EXPRESS-G.

C.2.2 Exchange structure with ISO 10303-201 or ISO 10303-202 as sheet

When this part of ISO 10303 identifies a drawing and ISO 10303-201 or ISO 10303-202 identifies sheets of the drawing, the AIM constructs in Figure C.2 shall be used in the implementation. Using this implementation method for this part of ISO 10303 utilizes constructs in ISO 10303-505.

The ISO 10303-505 constructs that are utilized in Figure C.2 are the following:

— **drawing_sheet_revision**.

One of the product_definition_formation constructs in Figure C.2 identify the drawing. The **drawing_-sheet_revision** construct and the other product_defintition_formation in Figure C.2 identifies the sheet in the drawing.

NOTE The graphical representation in Figure C.2 is presented in EXPRESS-G.

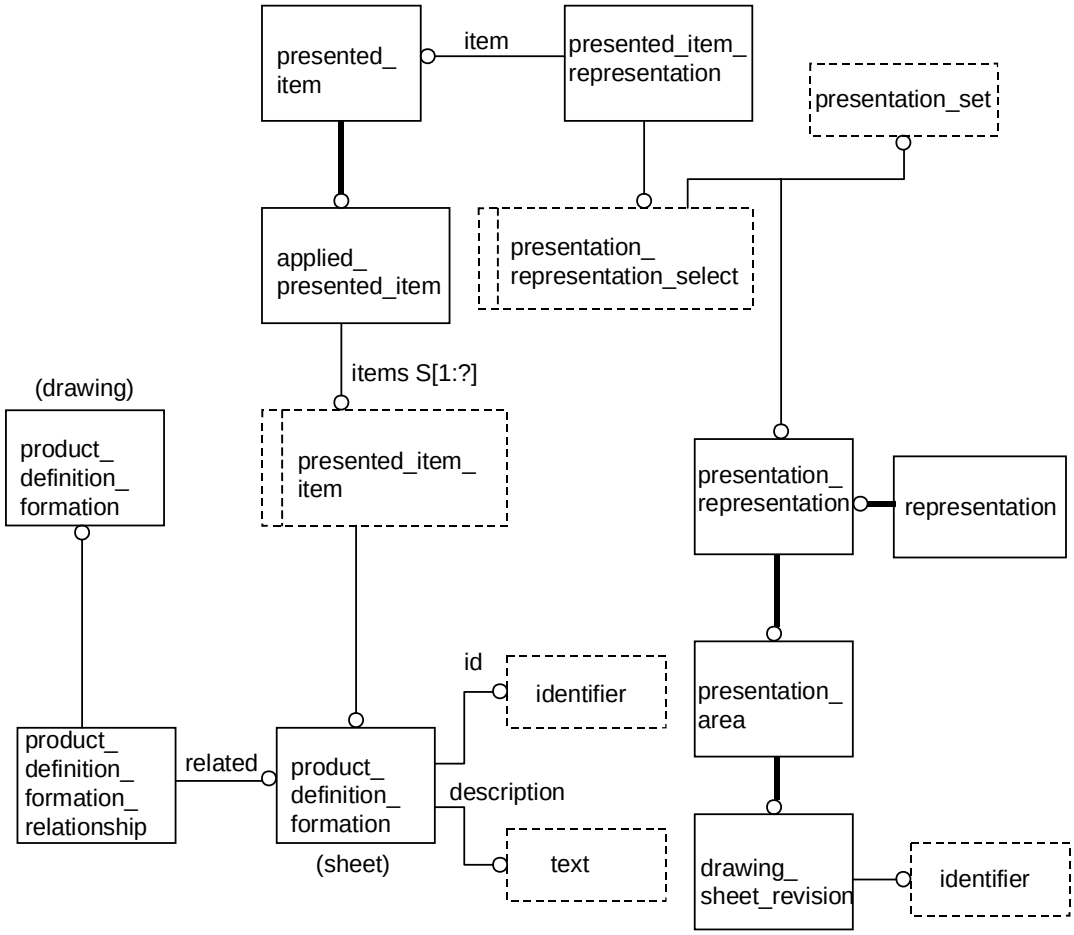

Figure C.2 — Exchange structure with ISO 10303-505 as sheet

C.2.3 Exchange structure with ISO 10303-201 or ISO 10303-202 as drawing

When ISO 10303-201 or ISO 10303-202 identifies a drawing and this part of ISO 10303 identifies sheets of that drawing, the AIM constructs in Figure C.3 shall be used in the implementation. Using this implementation method for this part of ISO 10303 utilizes constructs in ISO 10303-505.

The ISO 10303-505 constructs that are utilized in Figure C.3 are the following:

— **draughting_drawing_revision**;

— **drawing_version**.

The **draughting_drawing_revision** and **product_definition_formation** constructs in Figure C.3 identify the drawing. An other **product_definition_formation** construct in Figure C.3 identifies the sheet information in the drawing.

NOTE The graphical representation in Figure C.3 is presented in EXPRESS-G.

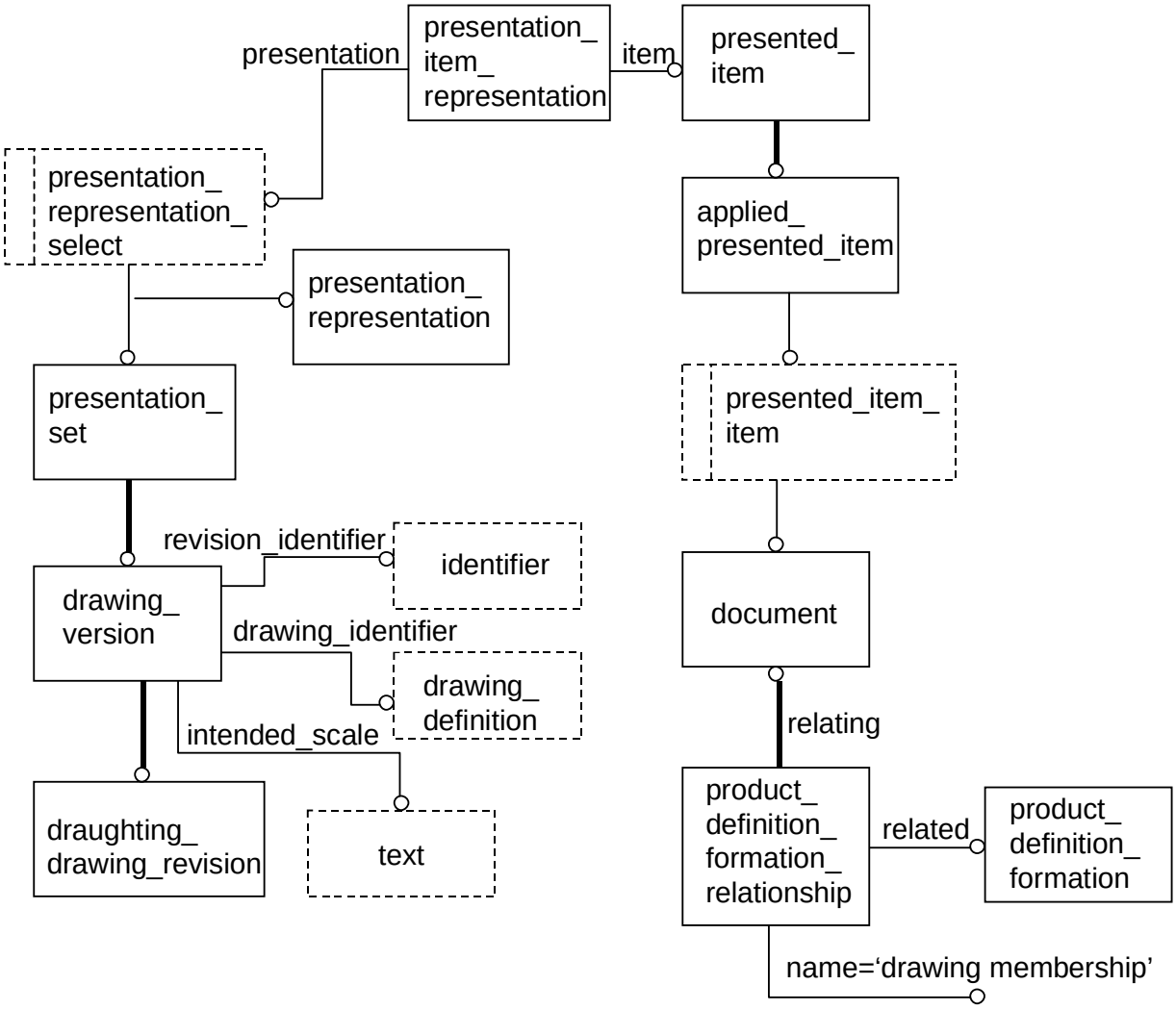

Figure C.3 — Exchange structure with ISO 10303-505 as drawing

C.3 Exchange structure for utilizing multiple confomance classes

Because all scopes of possible exchanges cannot be identified as individual distinct conformance classes, utilizing multiple conformace classes to satisfy a particular implementation may be required. An example of an implementation that needs multiple pieces of data that uniquely lie in different conformance classes is the combination of a Parts_list and a Product_data_set. The capability to associate a part's shape (geometry) within the context of an assembly is facilitated this way when the geometry is integrated with the same schema structure. Utilizing just the Part_list conformance class, the ability to reference an external file containing the shape geometry of the part is also available. With either of these part shape identification methods, digital mockup can be handled.

Annex D
(normative)

Protocol implementation conformance statement (PICS) proforma

This clause lists the optional elements of this part of ISO 10303. An implementation may choose to support any combination of these optional elements. However, certain combinations of options are likely to be implemented together. These combinations are called conformance classes and are described in this annex.

This annex is in the form of a questionnaire. This questionnaire is intended to be filled out by the implementor and may be used in preparation for conformance testing by a testing laboratory. The completed PICS proforma is referred to as a PICS.

Fourteen conformance classes are identified in this part of ISO 10303. A conforming implementation shall support at least one conformance class. Each class specifies a subset of ISO 10303-232 AIM constructs. These classes are detailed in clause 6 of ISO 10303-232.

Questions:

1. Please provide an identifier for the product or system for which conformance is claimed:

   Product name and current version number:_____

2. Please indicate the implementation method chosen:

   — ISO 10303-21 Exchange Structure -- preprocessor
   Preprocessor name and current version number:_____

   — ISO 10303-21 Exchange Structure -- postprocessor
   Postprocessor name and current version number:_____

3. Please indicate the classes for which conformance is claimed:

___ Class 1: Data definition exchange (DDE) for files;

___ Class 2: Data definition exchange (DDE) for technical data package elements;

___ Class 3: Data definition exchange (DDE) for indentured methods;

___ Class 4: Parts list;

___ Class 5: Data list;

___ Class 6: Indentured data list;

___ Class 7: Index list;

___ Class 8: Other list;

___ Class 9: List with presentation and the following:

- ___ Class 1: Data definition exchange (DDE) for files;
- ___ Class 2: Data definition exchange (DDE) for technical data package elements;
- ___ Class 3: Data definition exchange (DDE) for indentured methods;
- ___ Class 4: Parts list (PL);
- ___ Class 5: Data list (DL);
- ___ Class 6: Indentured data list (IDL);
- ___ Class 7: Index list (IL);
- ___ Class 8: Other list (OL);

___ Class 10: Reference document identification and drawing identification;

___ Class 11: Reference document identification and drawing identification with ISO 10303-201 and ISO 10303-202 drawing presentation identification;

___ Class 12: Product data set (PDS) without presentation format;

___ Class 13: Product data set (PDS) with shading;

___ Class 14: Product data set (PDS) with presentation format.

Annex E
(normative)

Information object registration

E.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(232) version(1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1 and is described in ISO 10303-1.

E.2 Schema identification

To provide for unambiguous identification of the schema specifications given in this application protocol technical_data_packaging in an open information system, object identifiers are assigned as follows:

{ iso standard 10303 part(232) version(1) object(1) technical-data-packaging-schema(1) }

is assigned to the technical_data_packaging expanded schema (see annex A);

{ iso standard 10303 part(232) version(1) object(1) technical-data-packaging-schema(2) }

is assigned to the technical_data_packaging short form schema (see 5.2).

The meaning of these values is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F
(informative)

Application activity model

The application activity model (AAM) is provided as an aid to understanding the scope and information requirements defined in this application protocol. The model is presented as a set of figures that contain the activity diagrams and a set of definitions of the activities and their data. The application activity model is given in Figures F.1 through F.17. Activities and data flows that are out of scope are marked with an asterisk for the definitions.

The viewpoint of the application activity model is from the engineer that has developed the Technical Data Package and from the engineer that must utilize the Technical Data Package in the operation and support of the product.

The original AAM [10] for this part of ISO 10303 was for an aerospace manufacturer composite structural part. From the original AAM, there has been refinement at the higher level to encompass the operations and support aspects of the product and some generalization at the lower levels to accommodate all technical data packages. The activities for an aerospace manufacturer are more detail and involve more supporting activities than most industrial practices for TDP preparation. Thus, the industrial TDP preparation practices will be accommodated.

Product Data Management (PDM) Systems product and document configuration activities are implied in many of the AAM activities. A key aspect of the scope of this part of ISO 10303 is to support configuration controlled exchange for technical data among PDM systems.

The AAM in this annex is focused on the activities that are used to create, format, and manage a product's full design and disclosure description, plus the activities that receive and utilize this information. Within this focus, this part of ISO 10303 will provide a link between ISO 10303-203 and ISO 10303-202. The ISO 10303-201 and ISO 10303-202 AAMs are focused on the act of capturing the geometric drawing without respect to what the product is. ISO 10303-203 is focused on capturing the configuration and geometric information of the product design without respect to the geometric drawing.

The AAM presented in ISO 10303-201 and ISO 10303-202 are representative of a designer who is required to document only the geometric aspects of a drawing and the annotation for that geometry. The ISO 10303--201 and ISO 10303-202 AAMs do not preclude the documentation of associated lists (for example, parts list, data list, and index list) information that is typically considered within a documentation package for a product, but they do not explicitly require the capture of that information nor do they capture that information in an intelligent manner. This part of ISO 10303 does capture associated lists in an intelligent computer interpretable manner, plus provides equivalent ISO 10303-201 and ISO 10303-202 capabilities, that are incoporated into an overall technical data package.

The AAM presented in ISO 10303-203 is also representative of a viewpoint from a designer. The ISO 10303-203 AAM highlights the capture of the product structure, the engineering change history, and the nominal shape of the product. However, ISO 10303-203 does not address tolerancing and drawing presentation (these are addressed in ISO10303-201 and ISO 10303-202).

This part of ISO 10303 will also capture the product structure, the engineering change history, and the nominal shape of a product, plus provide that information from a document configuration view.

F.1 Application activity model definitions

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions given in this annex do not supersede the definitions given in the main body of the text.

F.1.1
Acquire and Review Procurement Data
assimilates the technical aspects of the procurement data and reviews the technical aspects of the procurement data for completeness, consistency, currency, and applicability to a procurement of the Product

F.1.2
Analyze Requirements and Develop Modification
the analysis of the modification requirements, including field changes, and development of an initial modification for the Product plan for the Product.

NOTE This activity also determines if a modification satisfies the modification requirements and approves the modification for use in the field.

F.1.3 Anomaly Reports *
documentation analyzing the rejection of production parts for feedback to design and production planning

F.1.4
Apply Product Dimensions & Tolerances
the attachment of all the necessary dimensions and tolerances to the geometry of the drawing

F.1.5
Approved Modifications
the Product modifications that have been approved for manufacture and deployment to the field

F.1.6
Approved Modifications & Product Data
the Product modifications and Product data that has been approved for manufacture and deployment to the field

F.1.7
Approved Procurement Data
the technical aspects of the procurement data for a Product that has been approved for competitive procurement

F.1.8
Approved Product Data
the Product data that has been approved for manufacture of the modified Product

F.1.9
Approved Product Data List
list of product data that have been approved to be used in the list next life cycle step

F.1.10
Approved Product Reqs.
the functional and cross-functional requirements that have been reviewed and approved as being specific to a Product.

NOTE They include the technical performance constraints of the various functions along with the envelope features that are desired. The pertinent functions are from the Manage, Design, Build, and Support activities.

F.1.11
Archive Product Data*
the off-line storage of all product models, drawings, and referenced documentation for future use or as part of a disaster recovery program

F.1.12
Attach Product Engineering Notes
the attachment of all the Product engineering notes on the drawing

NOTE Engineering notes specify things such as process specifications, change notes, and material callouts.

F.1.13
Authorized Procurement Data
the technical and business procurement data that is authorized to be released for competitive procurement

F.1.14
Available Procurement Data
the procurement data that was available for a Product

F.1.15
Build and QA a Product*
the conversion of a design into a finished product and quality assurance functions that assure that the product meets requirements

NOTE This is usually an iterative function, continuing substantially throughout the product's life cycle. It receives the design from design functions and outputs the products, spare and repair parts, and technical data on each instance of the product.

F.1.16
Build Product Associated Lists
the building of the Associated Lists that accompany the Product design

EXAMPLE for a structural part - a parts list, a data list(s), an application list, indentured data list

F.1.17
Build Product Model & Drawing Tree
the building of a model/drawing tree that specifies the combinations of items used to create the Product

F.1.18
Changes due to Manufacturing Site Limitations
the changes to the Product information due to limitations of the Customer's manufacturing organization (for example, lack of equipment)

F.1.19
Changes due to Production
the changes to the Product information due to manufacturing limitations (for example, blemishes and schedule) that will still allow use of the Product

F.1.20
Changes due to Production Process Development
the changes to the Product information due to limitations of processes within the Customer's manufacturing organization (for example, high tolerance requirements)

F.1.21
Collect, Review, Define & Distr. Product Reqs.*
the collecting, reviewing, defining, and distributing structural, cross-functional engineering, build, QA, and logistic support requirements of the Product

F.1.22
Collected Product Baseline Data
all the referenced documentation (such as, specifications, standards, and other referenced documents) called out on the face of the drawing, along with all the required input geometry, system, and drawing data

F.1.23
Collected Product Baseline Drawing Data
all the drawing data from the selected Product preliminary design, test data, and producibility and maintainability studies

F.1.24
Collected Product Baseline Geometry
all the selected Product preliminary design geometry and other associated producibility and maintainability studies geometry

F.1.25
Collected Product Baseline Ref. Docs.
all the referenced documentation (such as, specifications, standards, and other referenced documents) called out on the face of the drawing

F.1.26
Collected Product Baseline System Data*
the standard features of the available CAD systems' hardware and software configurations

F.1.27
Company & Customer Review Stds
the documentation that ensures that the model has satisfied certain company and customer standards for quality and completeness

F.1.28
Competed Spares
the Product that will be competitively competed for manufacture

F.1.29
Conduct Product Prelim. Design & Analysis
the designing and anlysis of various Product concepts in order to trade performance, cost, and producibility parameters for selecting an optimum Product concept

F.1.30
Conduct Product Prototype Tests & Eval.*
all the physical and electronically simulated tests and evaluation of Product prototypes

F.1.31
Config Data
the configuration information of the product

F.1.32
Constraints of Selected Systems*
the limitations of the selected systems

EXAMPLE the limitations of data producing systems to output specific types of product data

F.1.33
Control Production*
plans the allocation of manufacturing resources to tasks defined in the manufacturing plan and produces detail shop authorization, including material, to perform the manufacture of the Product

F.1.34
Create Detail Product Design
the creation of the detail Product design based on inputs from the preliminary design phase and concurrent analyses

F.1.35
Create Product 2-D Layout*
the creation of the Product 2-D layout view geometry using conventional 2-D drawing entities within the selected construction planes

F.1.36
Create Product 3-D Wireframe*
the creation of the Product 3-D wireframe geometry using conventional 3-D drawing entities

F.1.37
Create Product Drawing Data
the creation of all the Product drawing data from the geometry and engineering specifications inputs using the selected systems

F.1.38
Create Product Informational Drawings
the creation of all supplemental drawings used in the development of the design

EXAMPLE logic diagrams or printed circuit board schematics, wiring diagrams, mechanical schematics, loading diagrams

NOTE These drawings may or may not be considered deliverable under the terms of the development contract and therefore may have varying degrees of formalness or drawing completeness

F.1.39
Create Product Layouts & Models
the creation of all of the necessary Product geometry layouts and models from the various inputs and prepare the data for transfer to other functions

F.1.40
Create Product Surface*
the creation of the Product surface geometry using conventional surface modeling entities

F.1.41
Create Product Solid Model*
the creation of the Product solid geometry using conventional or specialized solid entities

F.1.42
Data Lists
the listings of all engineering drawings, associated lists, specifications, standards, and subordinate data lists pertaining to the item to which the data list applies

F.1.43
Define Production Processes
the detail process steps and raw material requirements to manufacture a Product within the Customer's organization

F.1.44
Design and Analyze a Product
the complete design and analysis life cycle from the pre-proposal phase to product support in the field

F.1.45
Des. Reqs. Chgs. *
the changes from design activity resulting preliminary and detail design analysis

F.1.46
Design Staff & Tools*
personnel and tools from the contractor, the customer, and materials supplies

F.1.47
Det. Des. Reqs. Chgs.*
changes based on the analysis, producibility, and maintainability results

F.1.48
Detail Process Plan*
the detail process documentation of the process that is used to control the manufacture of a Product within the Customer's organization

EXAMPLE what types of machines for what processes

F.1.49
Detail Product Design Data
all the models, drawings, and parts list that make up the Product

F.1.50
Detail Product Item Drawings
all the design details of the subcomponents and assembly of the Product

F.1.51
Detail Product Test Reqs.*
the Product test requirements based on the design and analysis of the Product

F.1.52
Develop & Provide Product Materials*
the material suppliers' process of creating stock material for manufacturers

NOTE Basic material properties and allowables are addressed here.

F.1.53
Develop Modification Product Data
the development of the technical Product data for an initial prototype of the modified Product

NOTE This includes product data related to manufacture of the Product and to installation of the Product.

F.1.54
Develop Modifications
develop modifications due to excess field problems with the Product, with a change in requirements for use of the Product, field requested enhancements to the Product

F.1.55
Develop, Procure, Build, Use and Maintain a Product
the entire life cycle of a Government or Commercial Industry Product as viewed from the combined activity groupings of needs analysis and procurement; product development and fabrication by contractors, subcontractors, and raw material suppliers; and product use and maintenance

F.1.56
Develop Product Needs & Procurement*
the Customer's analysis of the Product requirements based on the customer's needs and the state of Product technologies, along with the procurement process throughout the life cycle as managed by the Customer

F.1.57
Develop Prototype
manufactures the prototype of the Product for testing

F.1.58
Dimensioned & Noted Product Details
the dimensioned, toleranced, and noted Product details as they would appear on a completed detail drawing

F.1.59
Dimensioned Product Details
the dimensioned and toleranced Product details as they would appear on the drawing

F.1.60
Dwg Prep Stds
documentation that determines how the Product models and administrative data are all presented in a standard format on a drawing

F.1.61
Exchange Prelim. Design Data*
exchanging preliminary design data with development partners, customers, and other off-site facilities for the purpose of reviewing developed design data or off-loading design tasks

F.1.62
Exchange Pre-Release Detail Design Data*
exchanging pre-release detail design data with development partners, customers, and other off-site facilities for the purpose of reviewing developed design data, off-loading design tasks, or exchanging the data as a deliverable product

F.1.63
Exchange Post-Release Detail Design Data
exchanging post-release detail design data with development partners, customers, and other off-site facilities for the purpose of reviewing developed design data or exchanging the data as a deliverable product

NOTE The data may be used to document/qualify the configuration of the product or used to support logistics functions.

F.1.64
Field Maintenance Changes & Revision History
the reviewed field/maintenance changes that result from in-field use of the Product and the history of these changes in the field

F.1.65
Field/Maint. Changes
the reviewed field/maintenance changes that result from in-field use of the Product

F.1.66
Field Changes*
the reviewed field changes that result from in-field use of the Product

F.1.67
Indentured Data List
the listing of all the engineering drawings, associated lists, and referenced documents pertaining to the level of product data package which applies

F.1.68
Field Product
the physical product that has been used in its field of operation

NOTE The product may or may not be usable.

F.1.69
Index List
the listing of all the data lists pertaining to the level of product data package which applies

F.1.70
Integrate & Prepare Product Assembly Drawings
integrate and prepare all of the items that make up the Product into an integrated assembly drawing

F.1.71
List Prep. Stds.
the national and company development standards, or the contractual requirements for list development, that define and regulate list format and content

F.1.72
Local Manufacture
the support activity for manufacture of additional Product within the Customer's organization

F.1.73
Maintain Native Format Detail Design Data*
sustain the detail design data elements developed for a product in the native systems in which the data was developed

NOTE The data is maintained in these systems for use by users and automated applications until the time when it is no longer useful or cost-effective to keep the data in the system.

F.1.74
Manage a Product Development*
managing all the resources specific to the Product through the design, build, and support functions

NOTE This includes managing things such as people, budgets, tools,and materials.

F.1.75
Manage Config. of Product Data*
the configuration management of all the data produced in the design development of the Product

F.1.76
Manage, Design, Build and Support a Product
the resource management, designing, building, and supporting of a typical Product, as done at the prime contracting company

F.1.77
Manufacturing Authorization*
the authorization to proceed in the manufacture of a Product

NOTE Manufacturing Authorization includes things such as work order information, quantity, and schedule.

F.1.78
Manufacturing Plan*
the overall Customer organization plan for manufacture of a product

F.1.79
Material Properties
the properties of the Product material that are used in the development of the modified Product Data

F.1.80
Material Specs & Reqs
definition of material composition, form, and performance requirements, along with orders procuring the material

F.1.81
Missing Procurement Data
the technical aspects of the procurement data that are missing and have to be present for manufacture of a Product

F.1.82
Model Prep Stds
documentation that describes how the Product models are all prepared to a minimum standard for the preparation of the geometry and associated engineering notes for the Product

F.1.83
Model & Dwg. Prep. Stds*
documentation that describes how the Product Models and Drawings are all prepared to a minimum standard for the preparation of the geometry and associated engineering notes for the Product

F.1.84
Modification Installation Plan*
the plan for installation of the modified prototype Product

F.1.85
Modified Product Test Results*
the results of the installation and testing of the modified prototype Product

F.1.86
Modification Test Plan Requirements*
the requirements for the testing of the Product

F.1.87
Modified Procurement Data
the technical aspects of the procurement data that were modified due to repair, modification, or change in requirements

F.1.88
Modified Product *
the Product that has been modified due to repair, change in requirements, or request for Product enhancement from the field

F.1.89
Modified Product Data
the Product data that has been modified for manufacture of a modified Product prototype

F.1.90
Mods
collection of information that is used to facilitate the creation of a modification to the Product

F.1.91
Mods to be Manufactured Locally
modifications to the Product that are to be manufactured within the Customer's organization

F.1.92
Mods to be Procured
modifications to the Product that are to be procured through competitive procurement

F.1.93
Non-Competed Spares
the identification of the Product that will be manufactured within the Customer's organization and will not be competitively competed for manufacture

F.1.94
Operate Product
the operation of the Product in the field

F.1.95
Other Lists
other lists that are prepared as part of the product data package

EXAMPLE wire lists, route lists, application lists.

F.1.96
Parts List
documentation listing the parts and bulk materials (except those materials that support a process) used in the item

F.1.97
Perform Cross-Funct. Product Reviews & CDR Functions*
the performance of the necessary cross-functional and customer design reviews to support the critical design review phase

F.1.98
Perform Product Detail Design & Analysis
creating and evaluating the Product design description and documenting it in sufficient detail to meet the desired performance, cost, and production goals

F.1.99
Perform Production Design & Anal. Support
testing the selected preliminary Product design concept and developing it in sufficient detail to meet the desired performance, cost, and production goods

F.1.100
Perform Production Processes
performs the actual manufacture of the Product within the Customer's organization

F.1.101
Perform Repairs
repairing the Product based on usage of the product in the field

F.1.102
Plan for Product Operation and Support
the planning of operation and support of the Product in the field

F.1.103
Pre-Rel. Product Design Data
the pre-released production support design and analysis data of the Product

F.1.104
Pre-Rel. Product Prelim. Data
the Product preliminary design and analysis data befre it is formally released by the owning organization

F.1.105
Prelim. Des. Reqs. Chgs.*
documentation of the preliminary design requirements changes based on the analysis, producibility, and maintainability results

F.1.106
Prelim. Funct. Reqs.*
the preliminary performance criteria the Product must meet

F.1.107
Prelim. Product Test Reqs.
the preliminary Product test requirements based on the preliminary design and analysis of the Product

F.1.108
Prepare and Collect Baseline Product Design Data*
the development of all necessary referenced documentation not already in existence and the collection of the referenced documentation and all other baseline design information required to develop a design of the product

EXAMPLE standard part drawings, material and process specifications, company fabrication standards, the selected preliminary design, test data, producibility, maintainability studies.

F.1.109
Prepare & Coordinate Signature Process
coordination of the distributing and collection of drawings that are to be reviewed and signed off by each reviewer

F.1.110
Prepare & Release Product AMRs*
the preparation of the Advanced Material Requests (AMR)s the engineering function sp long lead time material will be in inventory when needed

F.1.111
Prepare Data Lists
the development of a tabulation of all engineering drawings, associated lists, specifications, standards, and subordinate data lists pertaining to the item to which the data list applies and essential in-house documents necessary to meet the technical design disclosure requirements except for those in-house documents referenced parenthetically

F.1.112
Prepare Detail Product Item Drawings
creation of the drawing that contains the detail description of a product

F.1.113
Prepare Indentured Data List
tabulates all the associated lists and reference documents for a given drawing

NOTE As an alternative to the preparation of Data Lists and an Index List, an Indentured Data List may be prepared. Optionally, the Indentured Data List may tabulate all the associated lists and referenced documents for each part number identified within a given drawing. As with the index list, the Indentured Data List may reflect a listing of a single product data package or several product data packages.

F.1.114
Prepare Index List
the preparation of a tabulation of the data lists

NOTE The index list may reflect a listing of a single product data package (such as, a drawing and its immediate associated data/documents) or several product data packages. Other index lists may be prepared. However index lists that reflect tabulations other than data lists are prepared and treated under Activity A223245, "Prepare Other Lists".

F.1.115
Prepare Parts Lists
the preparation of a tabulation of all parts and bulk materials (except those materials that support a process) used in the item

NOTE Referenced documents may also be tabulated on parts lists. Items listed on subordinate assembly parts lists or specified in a referenced document are not repeated in the using assembly parts list unless it is necessary to limit options permitted by the subordinate document. In-house documents for in-house use only, may be referenced parenthetically.

F.1.116
Prepare Other Lists
the preparation of other lists that are prepared as part of the product data package

NOTE preparation of other lists include things such as wire lists, route lists, and application lists.

F.1.117
Prepare Procurement Package
assimilates the technical and business procurement data, coordinates with the Product procurement requirements, and prepares a procurement package for competitive procurement

F.1.118
Prepare Product Data for Use
archiving for future reference, converted to a transfer format in preparation for data exchange with downstream activities, or maintained in native format for reference by downstream activity usage & maintenance functions

F.1.119
Prepare Product Details
the preparation of the Product details to resolve the interfaces, joints, size, and the development of the detail part drawings

F.1.120
Prepare Product Installation Drawings
the preparation of the other subassemblies or assemblies that the Product is used on are shown on specific installation drawings

F.1.121
Prepare Product Models & Drawings
the preparation of the Product models and drawings using the reviewed design inputs and creating the necessary outputs for other functional uses

F.1.122
Procure Spares and Manage Procurement
manages the process of competitive and non-competitive procurement of a Product and distribution of the additional Product to the field

F.1.123
Procured Product
the actual procured Product

F.1.124
Procured Products & Information
the as-built Products and Information as purchased from outside associates or subcontractors

NOTE It includes not only the procured products, but also their respective drawings and material information such as stock, and material properties.

F.1.125
Product*
the as-built Product development

F.1.126
Product 2-D Layout
the 2-axis general layout and spatial arrangement views of the Product

F.1.127
Product 3-D Wireframe
a 3-D view of the Product features using conventional 3-D modeling entities that show the edges

F.1.128
Product AMRs*
the Advanced Material Requests by engineering of Product components that will be in inventory for the build cycle

F.1.129
Product Analysis Constraints*
constraints or limitations based on the analysis done on the Product

EXAMPLE strength limits for structural parts

F.1.130
Product Analysis Dwg. Changes
the changes that are recommended due to the analysis is reflected as redline marks to the design drawings

F.1.131
Product As-Built Definition Package
the as-built configuration management data of the Product

F.1.132
Product Assembly Drawings
pictorial showing how the detail component items are positioned in an assembly

F.1.133
Product Associated Lists
lists consisting of component constituent, part relationship, and document relationship tabulations that help to define the Product

EXAMPLE Parts List, Application List, Data List

F.1.134
Product Budget & Schedule
the projected cost and time it will take to develop the Product

F.1.135
Product Build Changes
changes that occur during the production process of the Product

F.1.136
Product Bus. Mgmt. Data
all the budget, cost, schedule, and people data used in the Product development

NOTE This also includes information such as contract number and design activity.

F.1.137
Product Component Interface*
all the component interfaces that mate with the Product

F.1.138
Product Contract
the business arrangement for the Product development as received from the customer

F.1.139
Product Data Modification Requirements
the requirements for modification of the Product data in the development of a modification and subsequent use in a prototype modified Product

F.1.140
Product Design Data
all the configuration managed design and analysis information that occurs for preliminary, detail, and production support phases of the Product

F.1.141
Product Design History*
all the similar design activities that have created data that is similar to the Product

F.1.142
Product Details
all the design data required of the Product as a result of mating interfaces and joints, and internal design, and manufacturing considerations

NOTE This all takes the form of drawings and parts lists that are distributed to other user functions.

F.1.143
Product Dev. Info.
all technical design data, business management data, and logistics support data used in the design, fabrication, and logistics support of the Product

F.1.144
Product Dev. Standards
the documentation of the national and company product development standards that apply to the Product

F.1.145
Product Dev. & Maintenance Techniques*
documentation of processes and methods for Product Development, Maintenance, and Material Development

F.1.146
Product Development Techniques*
processes and methods involved in the design, analysis, and build procedures and specifications used within Industry

F.1.147
Product Drawings
all the necessary component and assembly drawings

NOTE They include informational, component, assembly, and installation drawings

F.1.148
Product Informational Drawings
supplemental drawings such as schematics and diagrams

F.1.149
Product Installation Drawings
drawings that show how the Product is installed in other assemblies

F.1.150
Product Layouts & Models
all the two dimensional geometry required of the design and the appropriate geometry required of the design.

F.1.151
Product Maintenance Techniques*
the Maintenance Techniques as practiced in the field

F.1.152
Product Material Allowables
the characteristics of representative material and shape combinations of items within a Product

EXAMPLE strength or fatigue properties for structural parts

F.1.153
Product Material Development Techniques*
processes and methods as practiced by the material suppliers

F.1.154
Product Material Properties
the properties of the material that is used in the manufacture of a Product

F.1.155
Product Model & Dwg. Tree
the hierarchical relationships of the drawings and models of the Product

F.1.156
Product Models & Drawings
all the geometry and associated engineering notes for the Product

F.1.157
Product Modification Requirements*
modification requirements as dictated from use of the Product in the field, repair, or support requirements

F.1.158
Product Plan*
the plan for operating and maintaining the product in the field

F.1.159
Product Prelim. Build System Descr.*
description showing the system view of the manufacturing activities and resources needed to support the production of the Product

F.1.160
Product Prelim. Bus. System Descr.*
description containing the system relationships of the budget, schedule, and costs of a preliminary Product

F.1.161
Product Prelim. Design Sys. Descr.
description showing the functional, geometrical, and fit-up as of a preliminary Product using a graphical/ textual system engineering language

F.1.162
Product Preliminary Logistics System Description*
description showing the system view of the relationship between the logistics engineering, reliability and maintainability, spares, and training systems of a preliminary Product design

F.1.163
Product Producibility & Log Data & Build Changes
the as-built configuration management data of the Product that must be incorporated into the documentation of the designed Product

F.1.164
Product Prototypes*
of all the pre-production physical or electronic models of the Product

F.1.165
Product Raw Material*
the raw material for the Product

NOTE The raw material may be another Product or may be stock material.

F.1.166
Product Raw Materials & Material Properties
all of the raw materials as received from the material supplier

NOTE These are for the item(s) of the Product. It includes the properties of the material as well as the rough material: stock.

F.1.167
Product Review Data*
data that has been or is about to be reviewed by the customer

F.1.168
Product Solid Model
the graphical representation of the Product using boundary representation (B-REP) or constructive solid geometry (CSG) elements

F.1.169
Product Surface
the graphical rendering of the boundary faces of the part using surface geometry elements

F.1.170
Product Spares*
Production Parts that have been procured or manufactured for logistical and maintenance support of a product

F.1.171
Product Support Logistics Data*
of all the logistics engineering, reliability and maintainability, technical and maintenance, documents, and spares data needed to support the Product

F.1.172
Product Support Logistics Data & Build Changes
all the logistics engineering, reliability and maintainability, technical and maintenance, documents, and spares data needed to support the Product

NOTE It also includes any required changes that must be incorporated into the documentation of the designed Product so that appropriate support can be achieved.

F.1.173
Product Test Data*
all of the test data from the verification tests of the product and its subcomponents

F.1.174
Production Part*
the final manufactured component, assembly, or product that has been produced

F.1.175
Production Status*
the status of shop machinery, personnel, schedule, and other data related to the shop floor completing the manufacture of the Product

F.1.176
Prototype Product*
the prototype product that is to be installed and tested for possible use in the field

F.1.177
Receive, Review, and Plan Manufacturing Requirements*
assimilates the manufacturing requirements and plans for manufacture of the Product within the Customer's organization

F.1.178
Reference Documentation
all specifications, national and company standards, and all other documents that can be referenced from the face of a drawing and are necessary for completing the Product's design disclosure

EXAMPLE Material and process specifications, drawing interpretation and fabrication practices, engineering instructions

F.1.179
Referenced Documents
all specifications, national and company standards, and all other documents that are referenced from the face of a drawing and are necessary for completing the Product's design disclosure

EXAMPLE Material and process specifications, drawing interpretation and fabrication practices, engineering instructions

F.1.180
Review Product Baseline Geom. Data & Select Construction Planes*
the review of all the different forms (paper, translated, native) of Product baseline geometry and select the Product construction planes that render the desired views of the Product for top, front, side, or cross-section details

F.1.181
Reviewed 2-D Baseline Geometry
the data that was received from design support sources that support the geometry development by the - Product

F.1.182
Reviewed 3-D Baseline Geometry
the data that was received from design support sources which support the geometry development by the - Product

F.1.183
Select & Detail Product Views
the selection and detailing of the necessary Product views based on the typical top, front, side, and cross-sections needed to show the desired features

F.1.184
Select & Prepare Model/Draft. System*
the selection and preparation of the modeling/drafting geometry system to be used for the detail design phase of the Product

F.1.185
Selected Dwg. Sys.*
the selected drawing system needed to support the detail design stage

F.1.186
Selected Mod./Dwg. Sys.*
the selected modeling system and drawing system needed to support the detail design stage

F.1.187
Selected Modeling Sys.*
the selected modeling system needed to support the detail design stage

F.1.188
Selected Product Construction Planes
the associated cartesian axes of the top, front, side, or cross-section views desired of the Product

F.1.189
Selected Product Prelim. Des.
the Product preliminary design selected from the various concepts that were evaluated against one another

F.1.190
Selected Product Views
the top, front, side, and cross-section views necessary to show the geometric features

F.1.191
Selected Sys. Constraints*
the inherent drafting or modeling technique constraints of the design toolset

F.1.192
Selected View Constraints
the construction planes of the top, front, side, or cross-sections that are location dependent

F.1.193
Shop Work Order*
the shop authorization to proceed with the manufacture of a Product

NOTE It includes things such as the detail process plan, quantity, and schedule.

F.1.194
Spares Procurement
the support activity for procurement of additional Product to support the operation of the Product in the field

F.1.195
Spares to be Manufactured
the identification of the Product and the quantity of product that is to be manufactured within the Customer's organization

F.1.196
Staff & Tools*
personnel and tools from the contractor, the customer, and materials supplier

F.1.197
Support Logistics of a Product*
logistics engineering, reliability and maintainability design studies, technical and maintenance documents, spares, and training systems that the Product repairs

F.1.198
Support Product
the support of the Product outside of the field activities

F.1.199
Tested Product Prototype*
all the pre-production physical or electronic models of the Product that have been tested

F.1.200
Update Product Drawing & Model Data*
the update of the Product drawings and models based on the changes to the Product

F.1.201
Updated Product Models & Drawings
the updated Product drawings and models that have incorporated the most recent design changes

F.1.202
Use & Maintain a Product
the Customer's use and maintenance of a Product

NOTE It also includes repair, redesign, and modification activities of a Product at the servicing location.

F.1.203
Useable Product*
the product in a useable condition

NOTE The product may have undergone revision, repair, retrofit, and/or maintenance. The product may be a spare.

F.1.204
Used Product*
the Product as a result of use in the field

F.1.205
Validate, Revise, and Store Procurement Data
validates that the technical aspects of the procurement data is valid for a particular procurement, modifies the technical aspects of the procurement data (if appropriate), stores the technical procurement data, determines if the procurement will be competitive or non-competitive, and approves the technical content of the procurement data for a Product

NOTE If data is missing or not useable, this activity initiates remedial action to acquire the required data.

F.2 Application activity model abbreviations

The following abbreviations are used in the diagrams:

AAM Application Activity Model;

AMR Advanced Material Requests;

Anal. Analysis;

AP Application Protocol;

Bus. Business;

CDR Critical Design Review;

Chgs. Changes;

Config. Configuration;

Des. Design;

Descr. Description;

Det. Detail;

Dev. Development;

Docs. Documents;

Dwg.	Drawing;
Eval.	Evaluation;
Funct.	Functional;
IML	Inner Mold Line;
Info.	Information;
Maint.	Maintenance;
Matl.	Material;
M/D/B/S	Manage/Design/Build/Support;
Mgmt.	Management;
Mod.	Modeling;
Mods.	Modifications;
M&P	Materials & Processes;
OML	Outer Mold Line;
PDES	Product Data Exchange using STEP;
Prelim.	Preliminary;
Prep.	Preparation;
Prop.	Property;
QA	Quality Assurance;
Rel.	Released;
Reqs.	Requirements;
Stds.	Standards;
STEP	STandard for the Exchange of Product model data;
Sys.	System;
TDP	Technical Data Package;
Tol.	Tolerance.

F.3 Application activity model diagrams

The application activity model diagrams are given in Figures F.1 through F.17. The graphical form of the application activity model is presented in the IDEF0 [11] activity modeling format. Activities and data flows that are out of scope are marked with a dashed box or a dashed line, respectively.

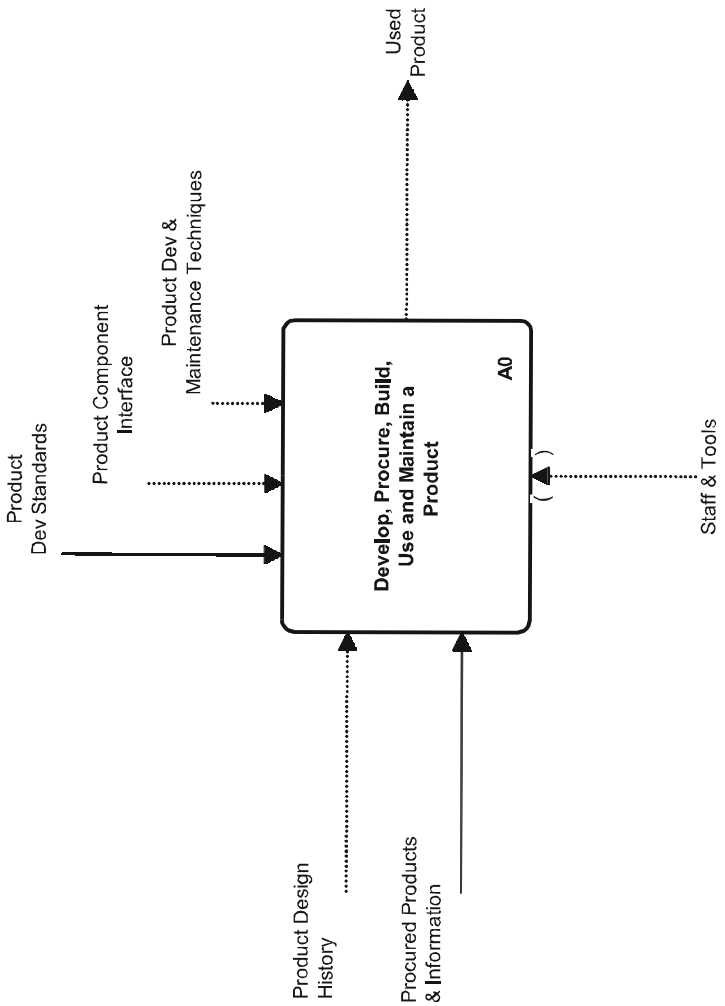

Figure F.1 — A-0: develop, procure, build, use, and maintain a product

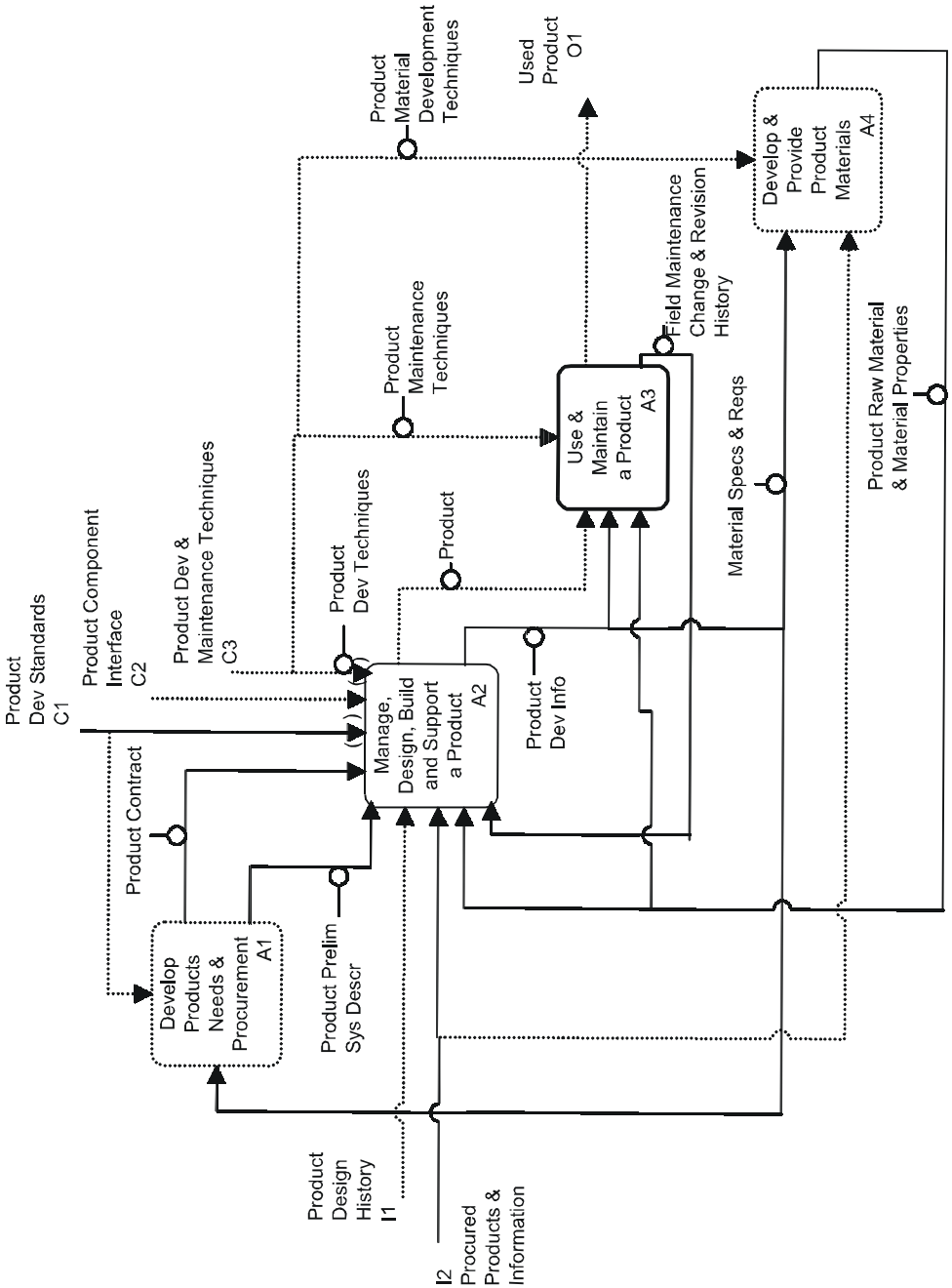

Figure F.2 — A0: develop, procure, build, use, and maintain a product

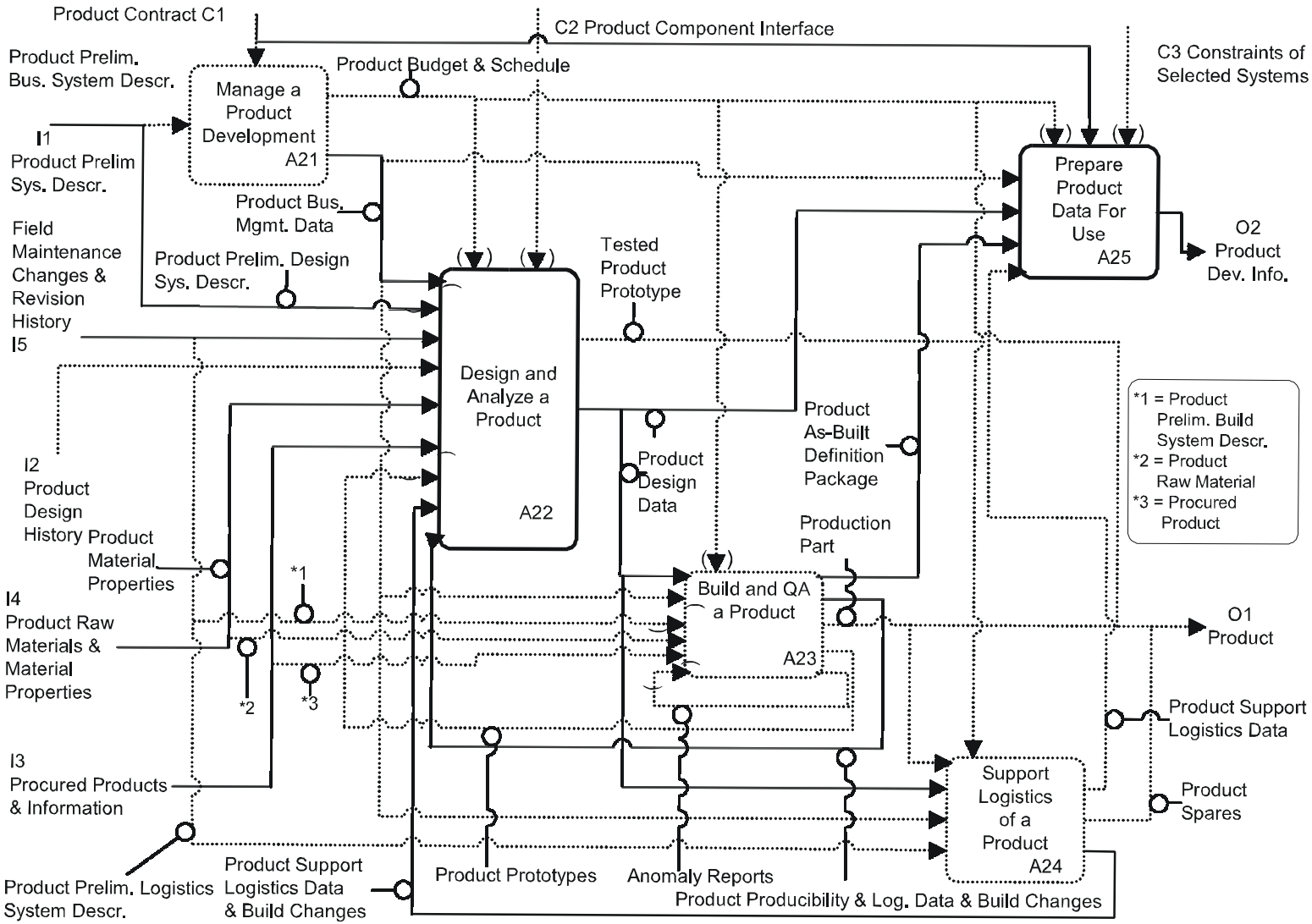

Figure F.3 — A2: manage, design, and support a product

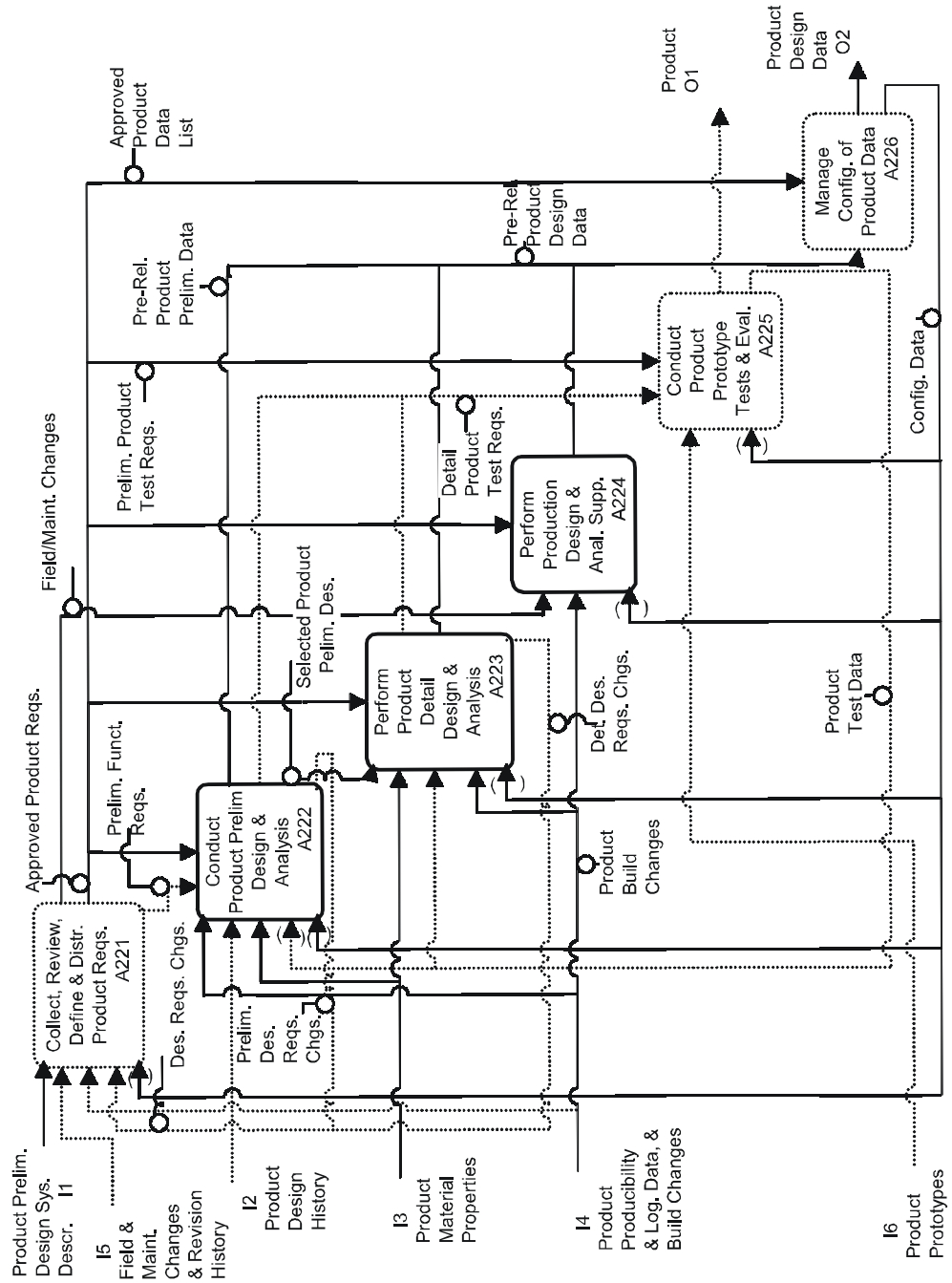

Figure F.4 — A22: design and analyze a product

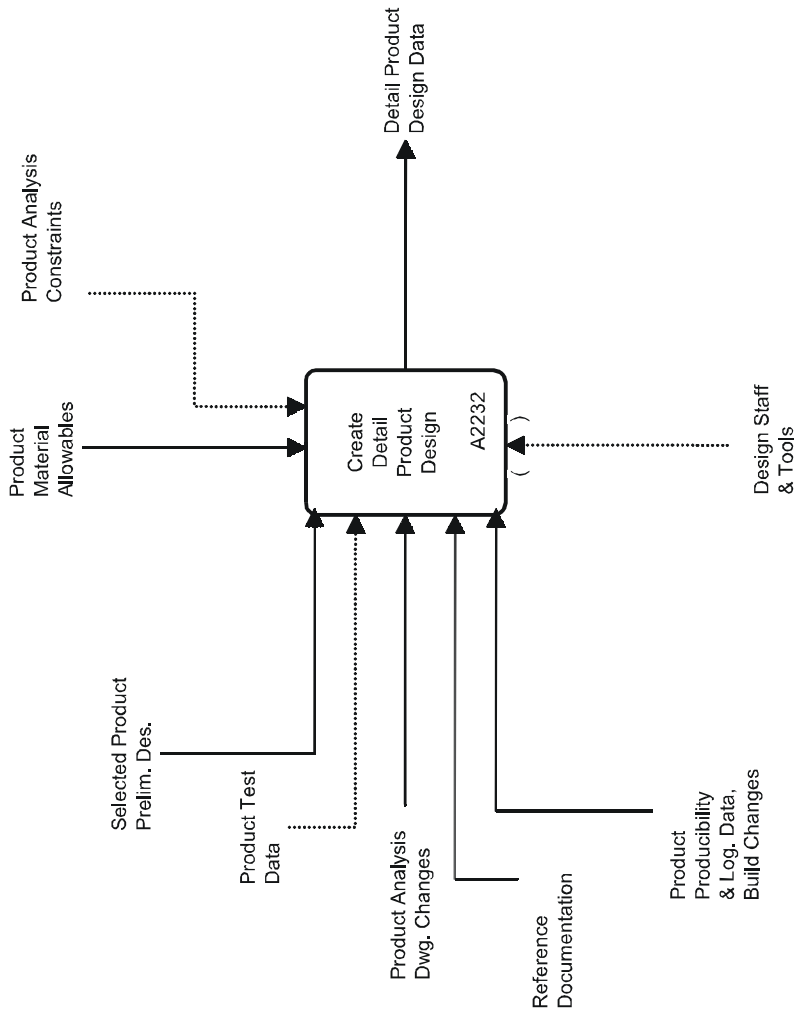

Figure F.5 — A223: perform product detail and design analysis

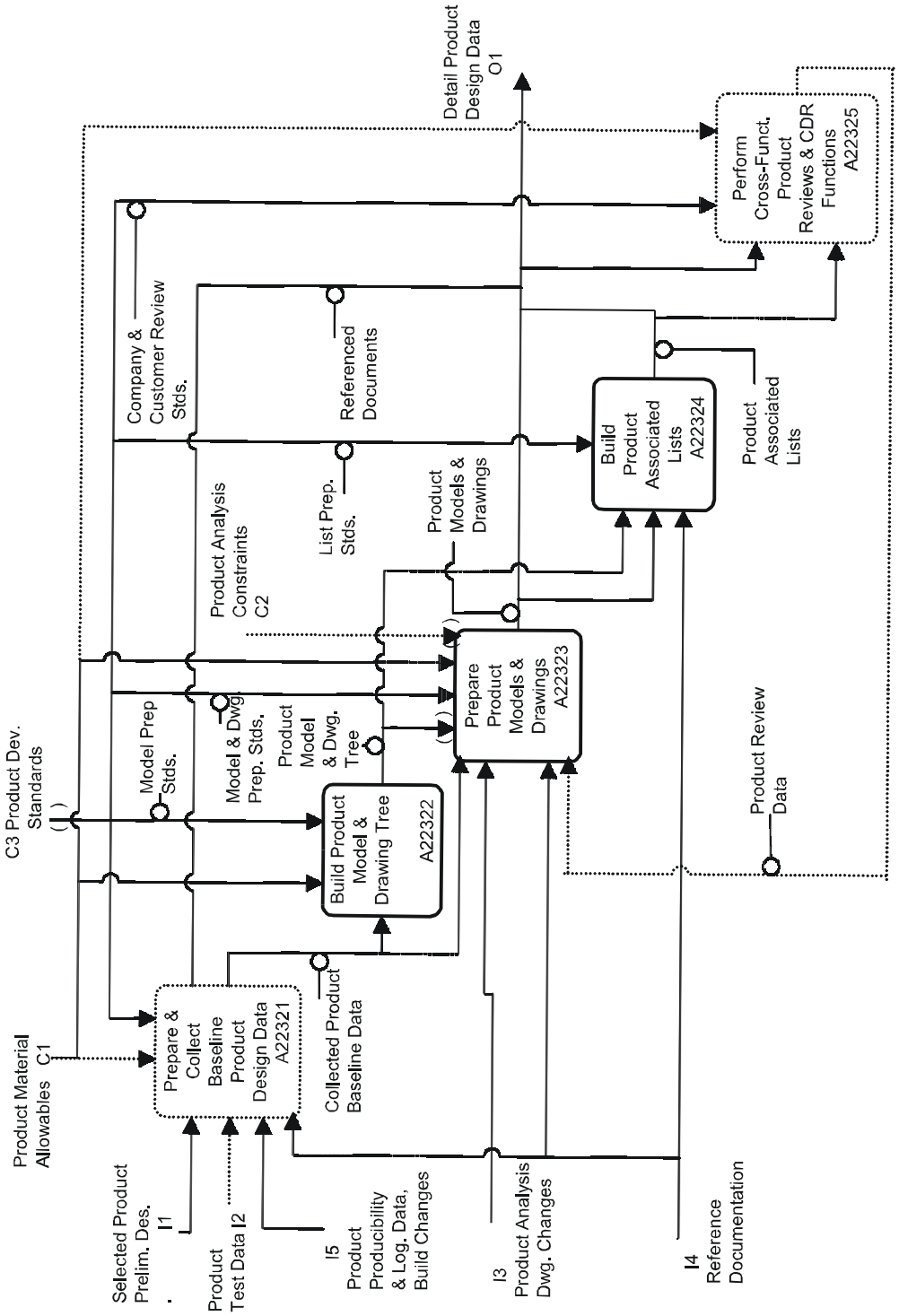

Figure F.6 — A2232: create detail product design

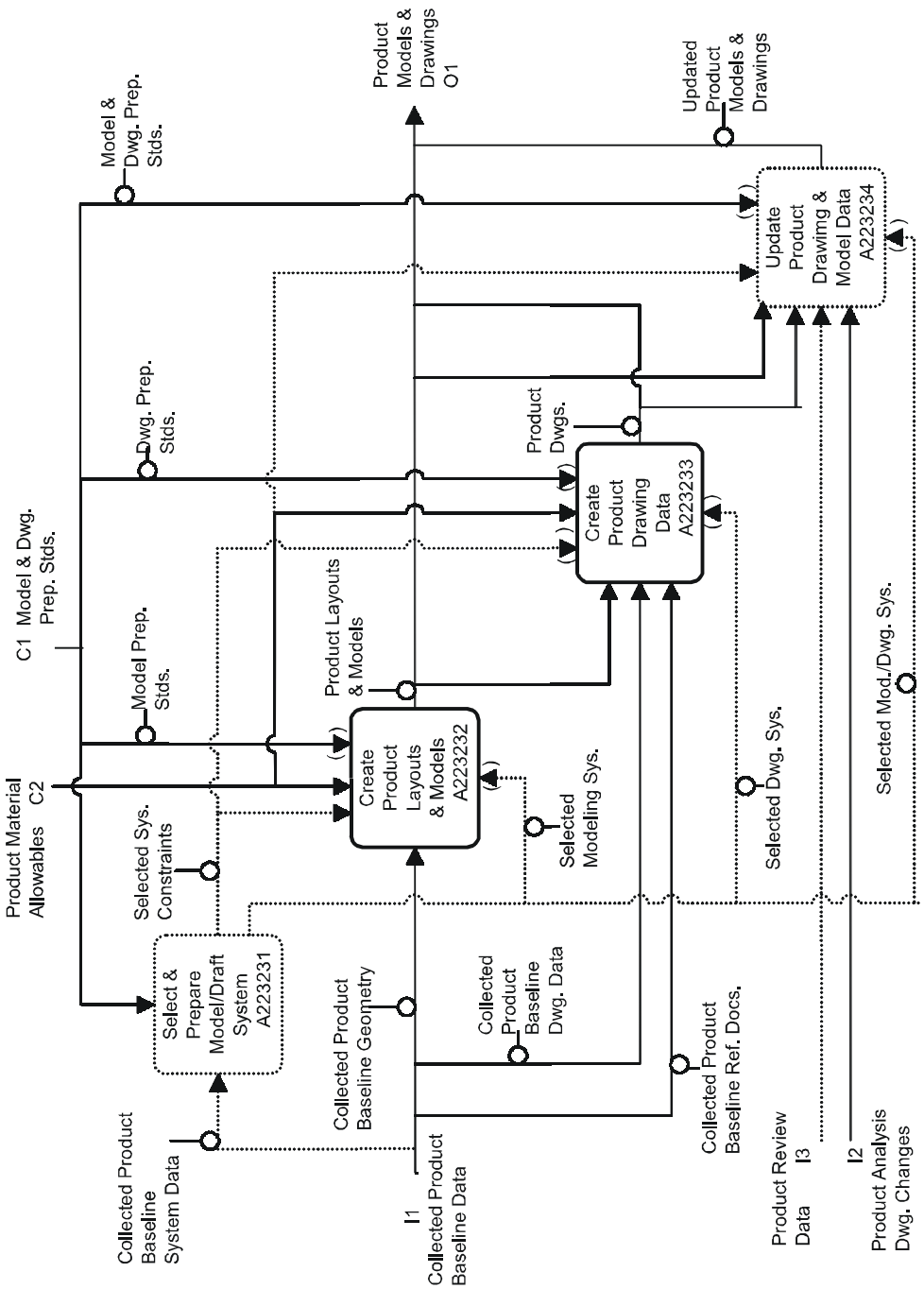

Figure F.7 — A22323: prepare product models and drawings

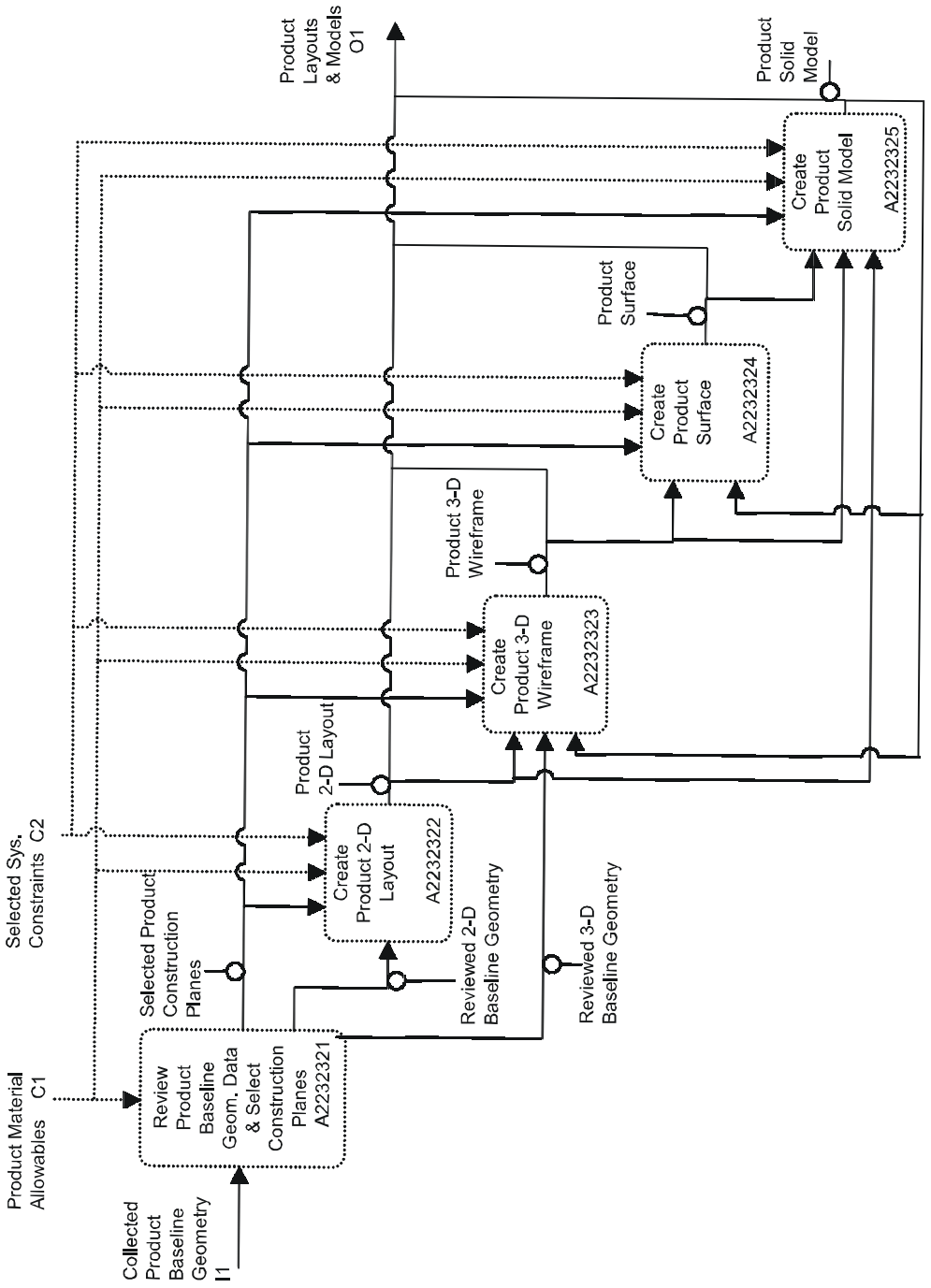

Figure F.8 — A223232: create product layouts and models

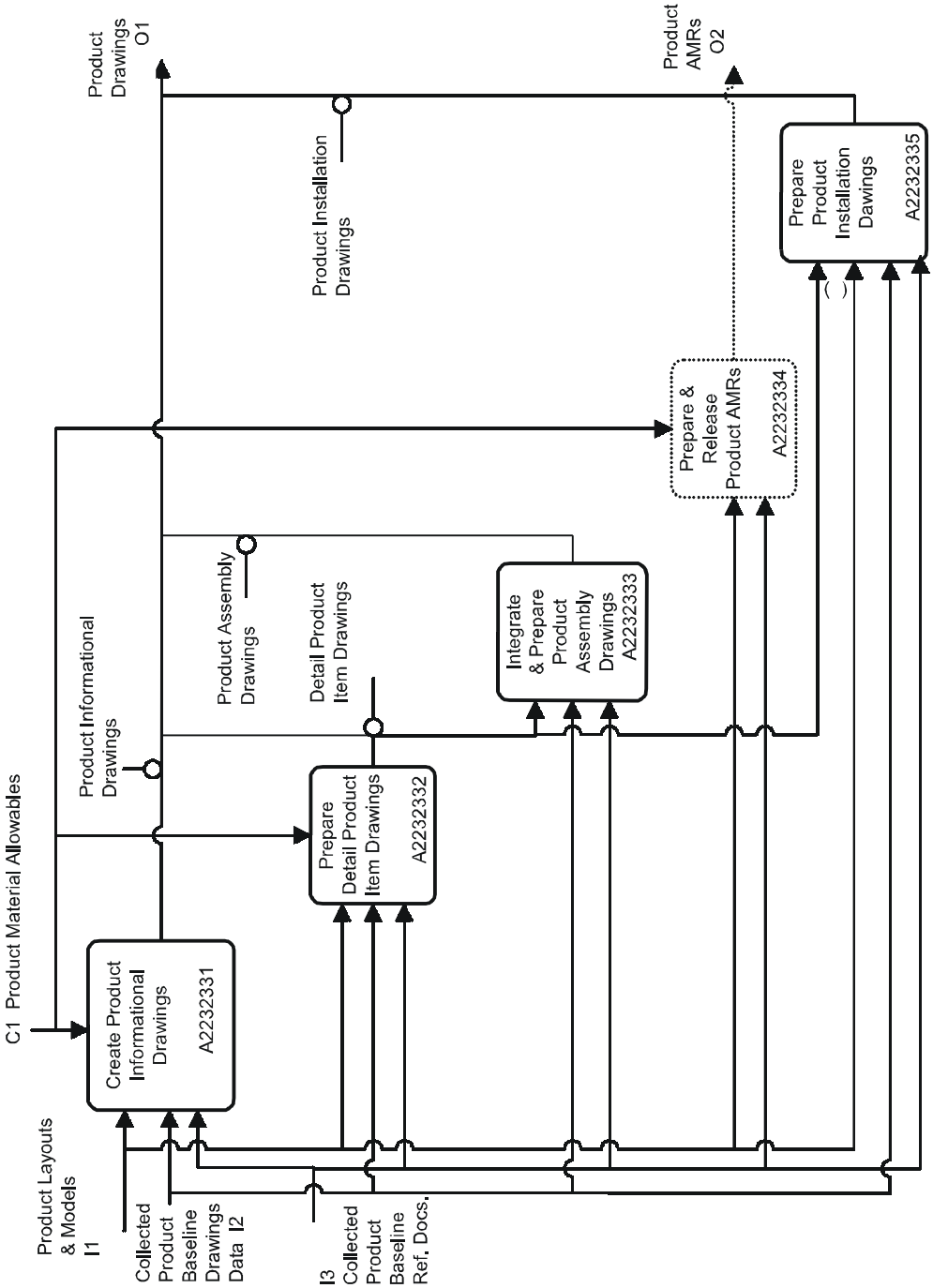

Figure F.9 — A223233: create product drawing data

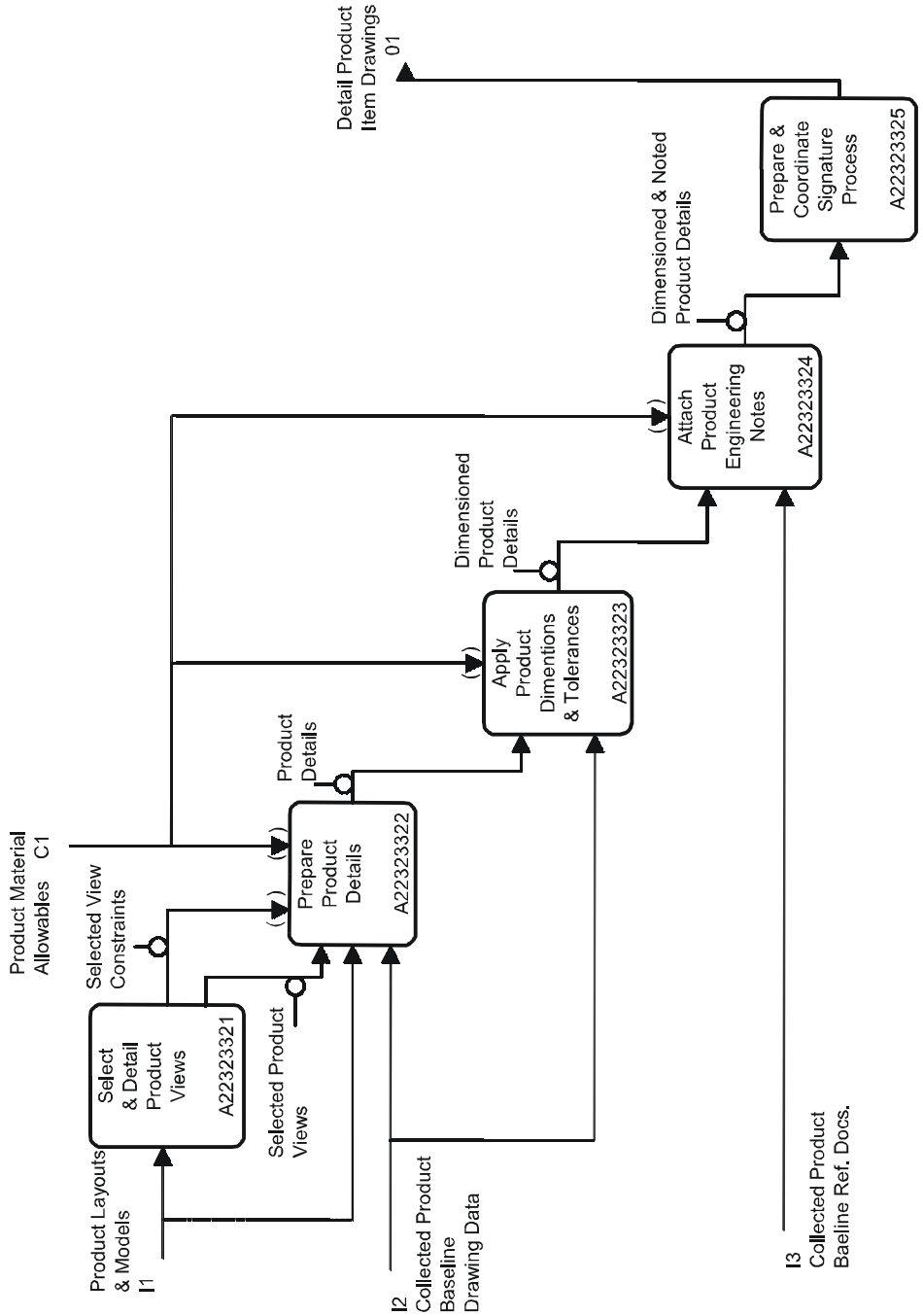

Figure F.10 — A2232332: prepare detail product item drawings

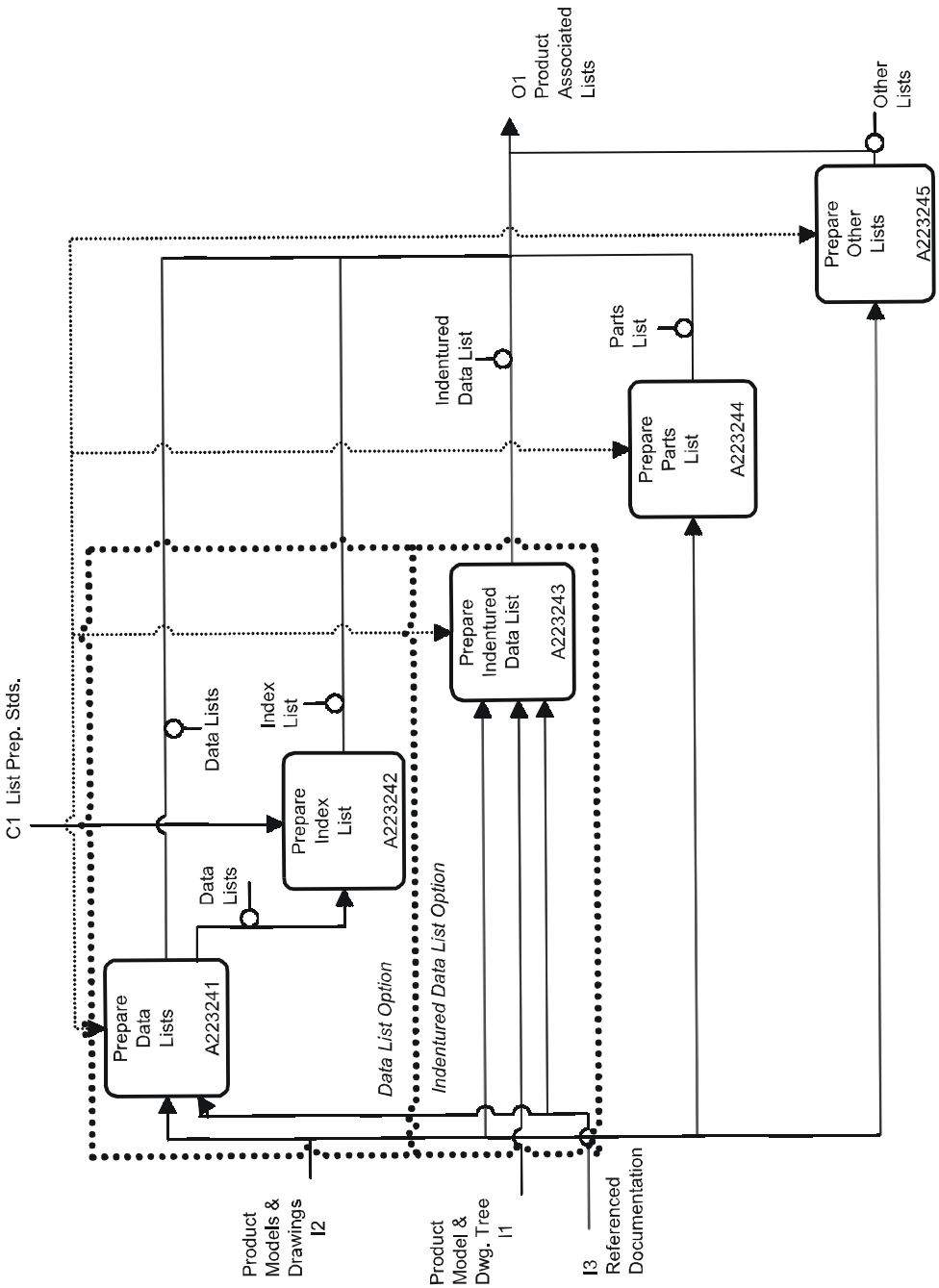

Figure F.11 — A22324: build product associated lists

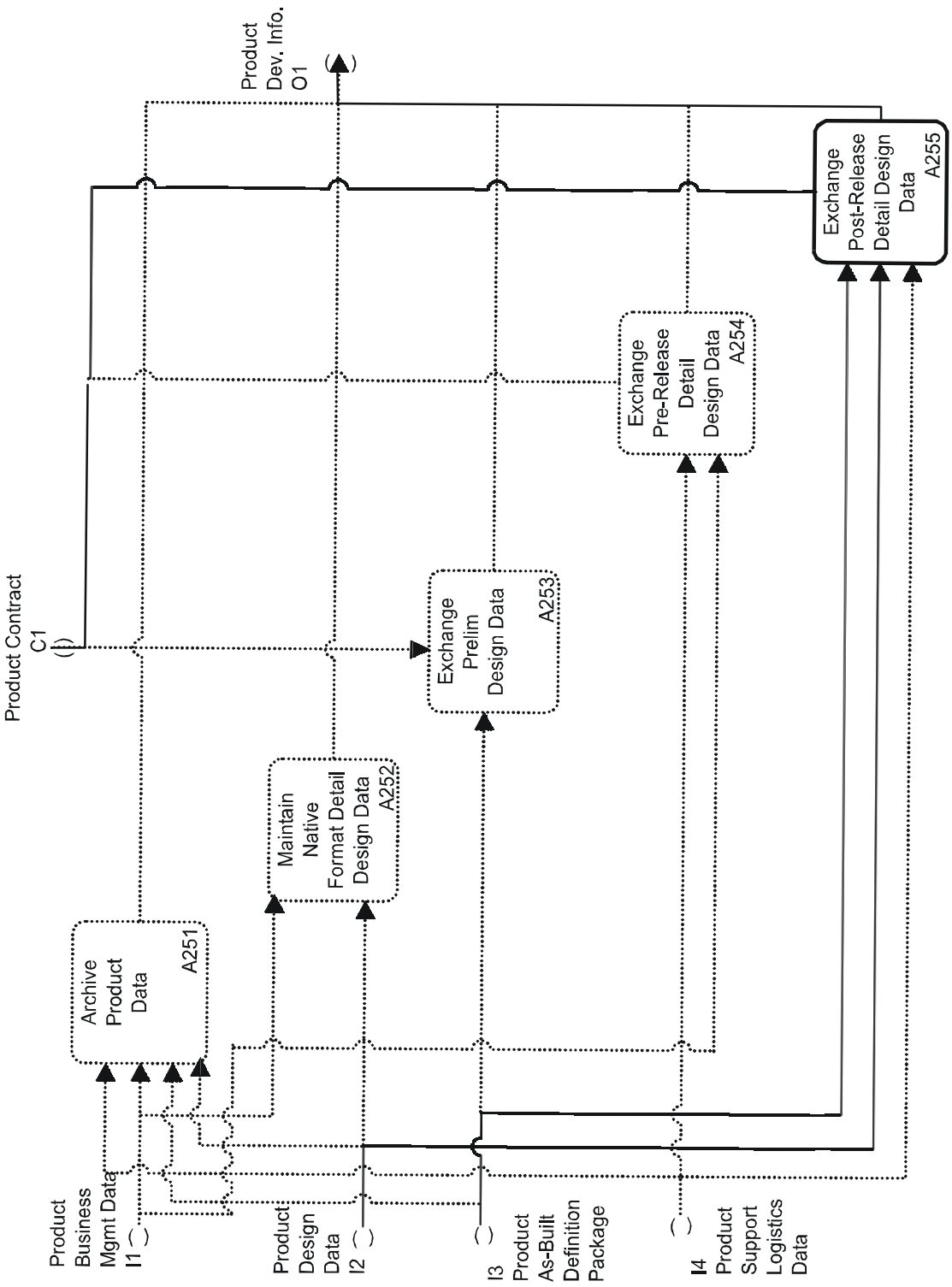

Figure F.12 — A24: prepare product data for use

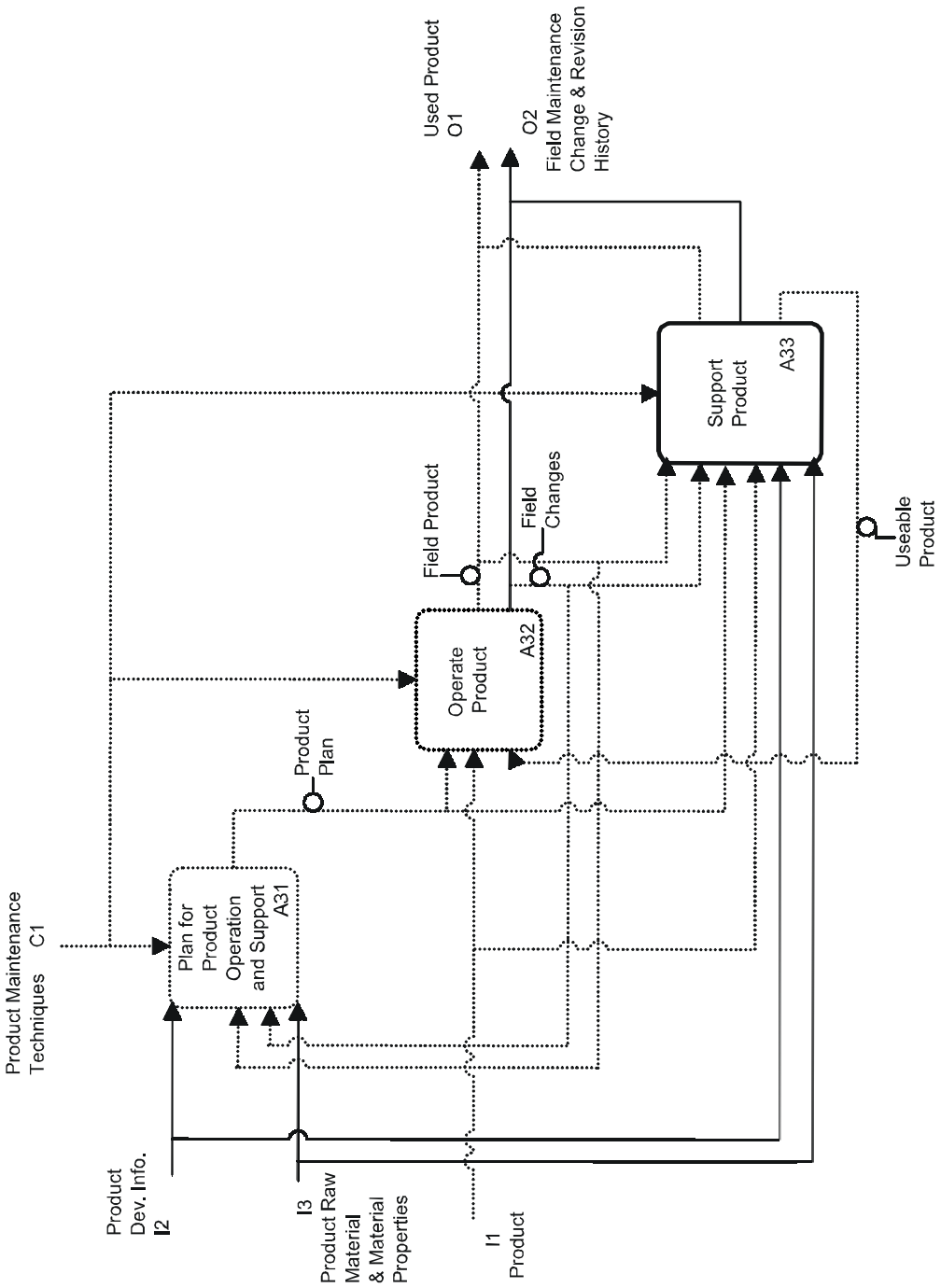

Figure F.13 — A3: use and maintain a product

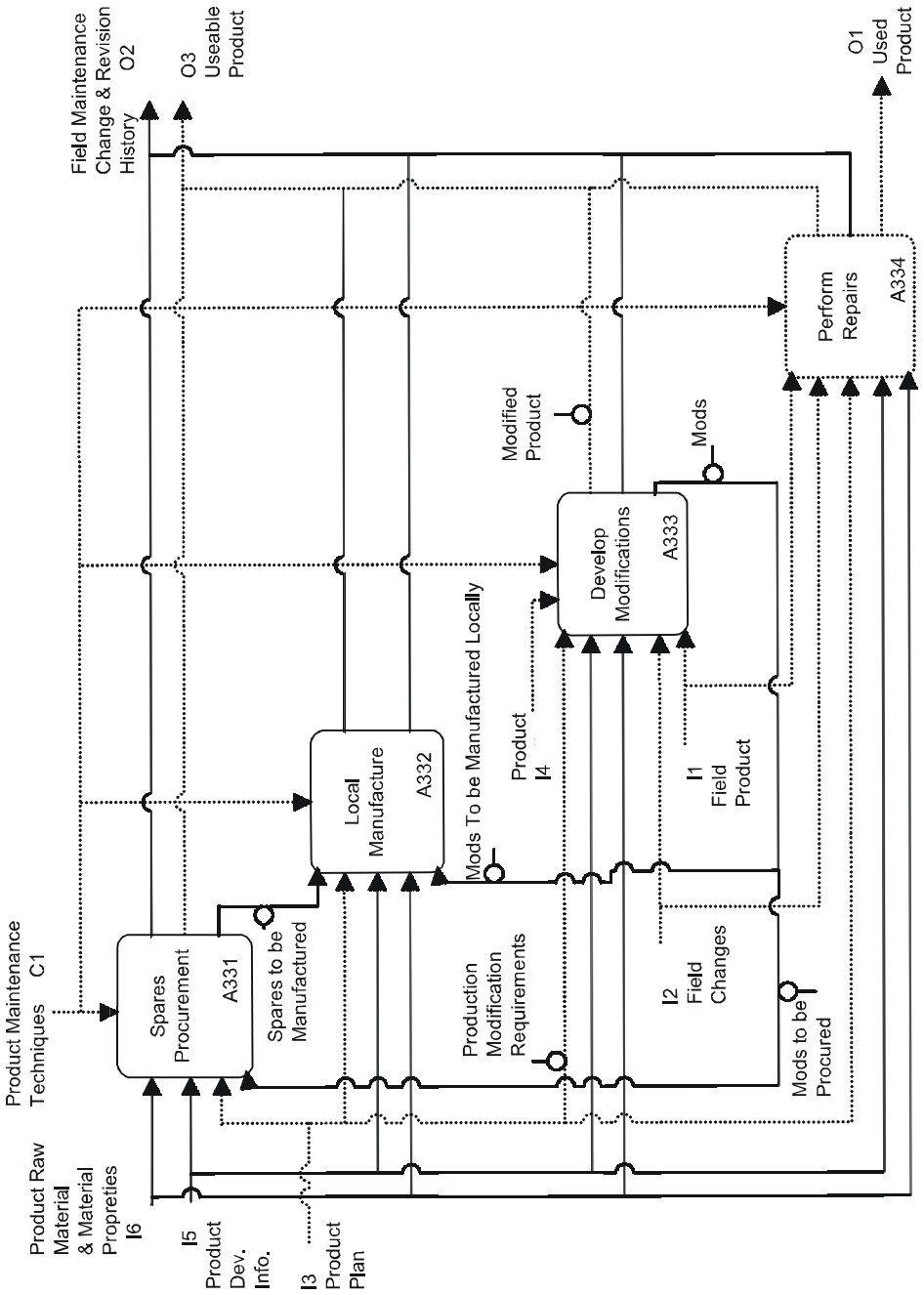

Figure F.14 — A33: support product

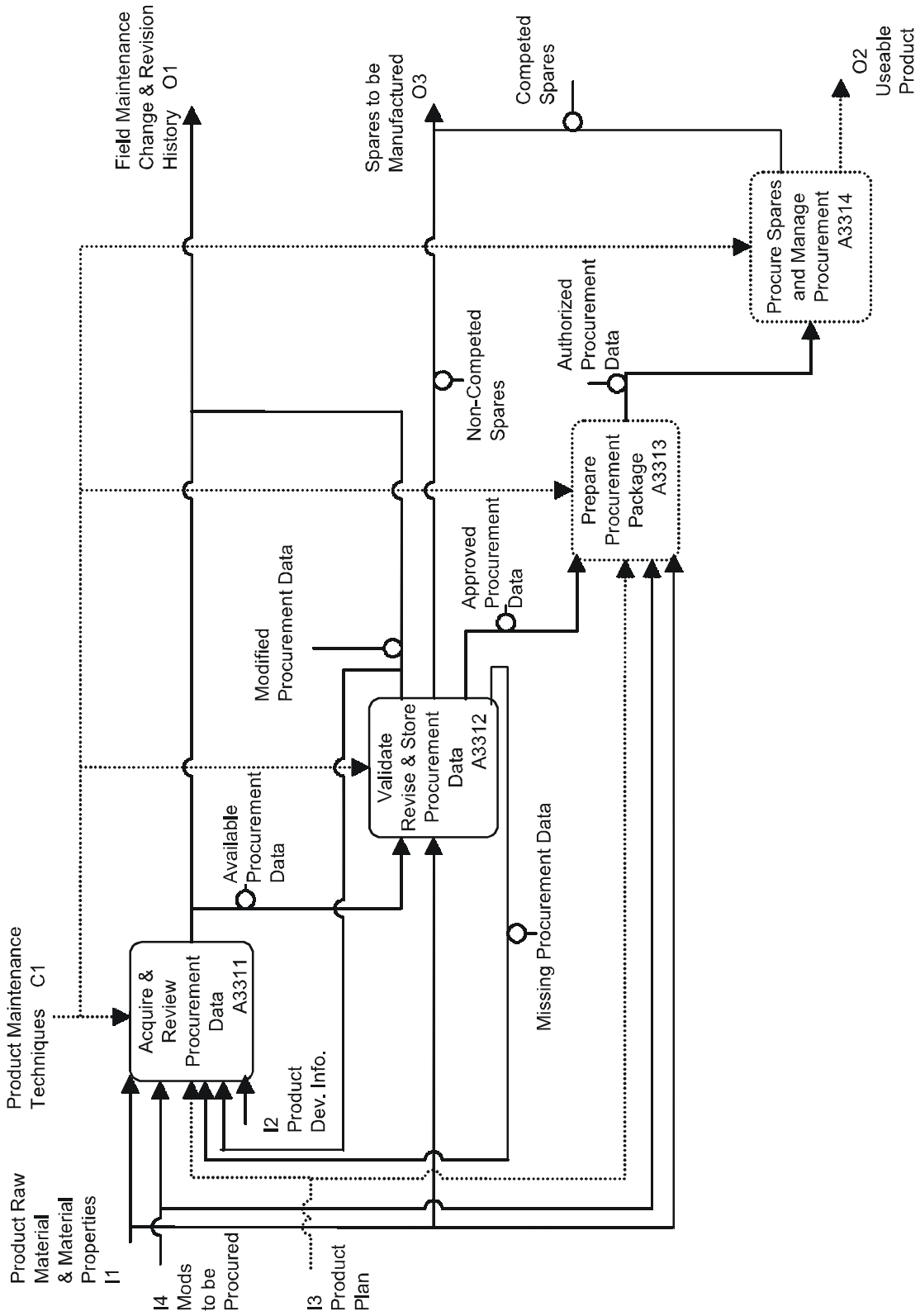

Figure F.15 — A331: spares procurement

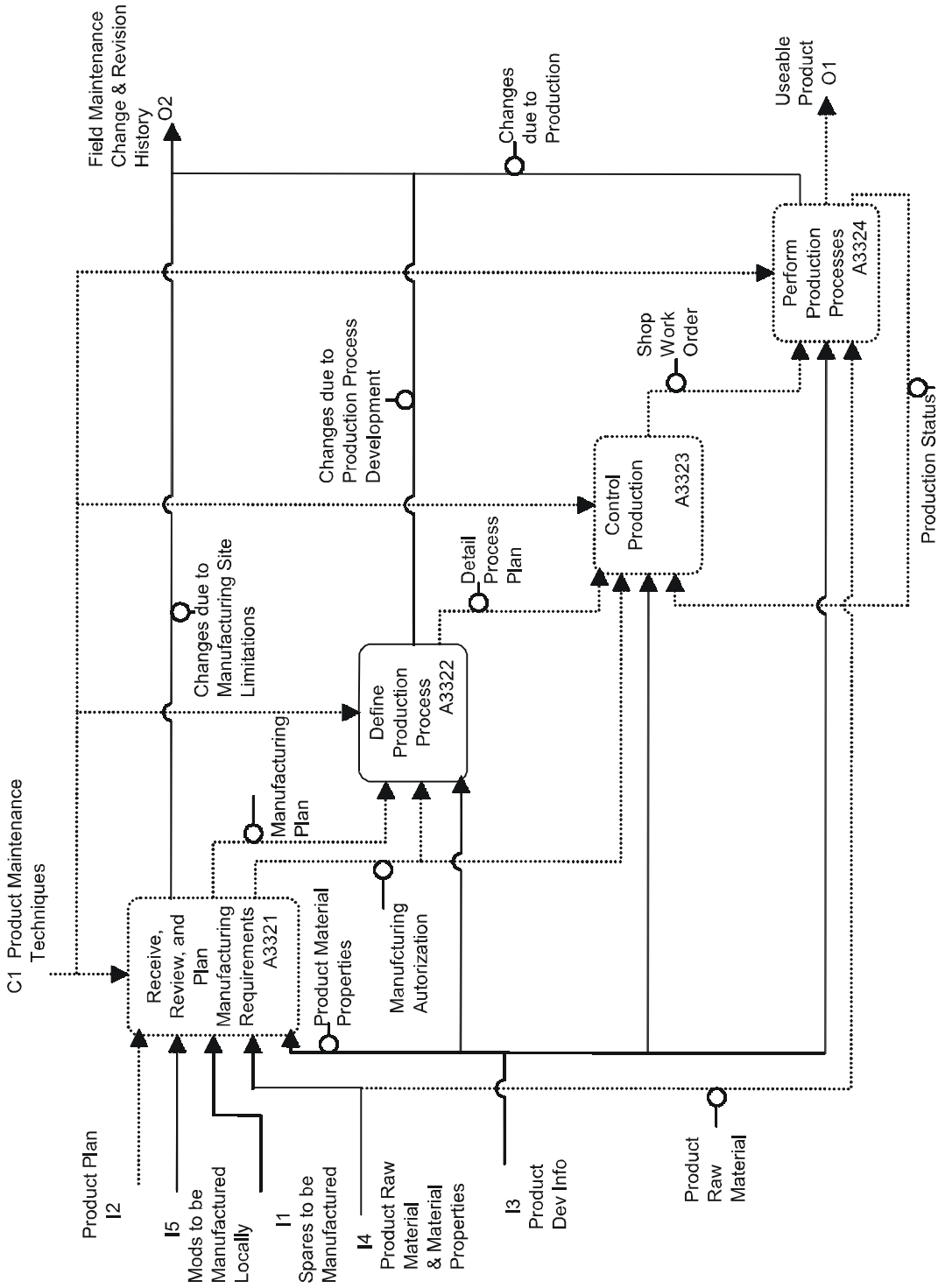

Figure F.16 — A223: local manufacture

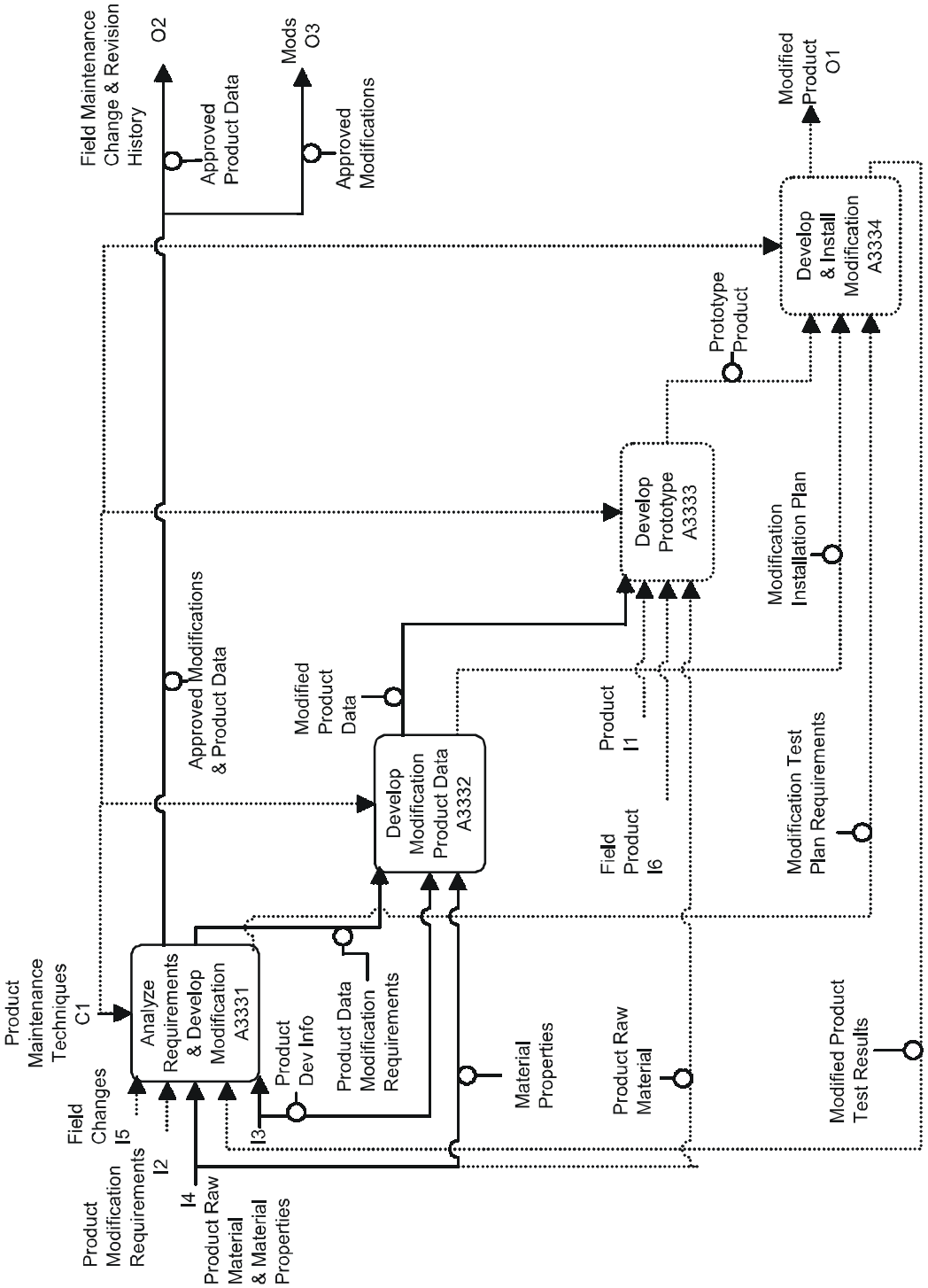

Figure F.17 — A333: develop modifications

Annex G
(informative)

Application reference model

This annex provides the application reference model for this part of ISO 10303 and is given in Figures G.1 through G.33. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in EXPRESS-G. EXPRESS-G is defined in annex D of ISO 10303-11. The application reference model is independent from any implementation method.

NOTE The following EXPRESS-G diagram listing may be used as an aid for locating objects within the application reference model Figures.

EXPRESS-G diagram index

TYPE Figure
a_number . . . G.32
a_real . . . G.32
an_integer . . . G.32
identifier . . . G.33
label . . . G.32
measure . . . G.32
text . . . G.33
yes_no . . . G.4

ENUMERATION_OF
change_type . . . G.19
component_list_type_enumeration . . . G.9
exchange_reason . . . G.2
quantity_accuracy_enumeration . . . G.21
retrofit_state . . . G.8
stock_size_classification . . . G.10
stock_size_cross_section . . . G.10
weight_derivation . . . G.16

SELECT_TYPE
alternate_identification_element_select . . . G.20
alternate_identification_item_select . . . G.20
component_list_type_select . . . G.9
data_definition_entry_and_file_select . . . G.3
data_definition_entry_select . . . G.3
data_definition_exchange_list_method . . . G.3
data_definition_file_entry . . . G.6
date_or_event G.22
drawing_or_product_data_set . . . G.29
entry_format_select . . . G.4
geometry_select . . . G.10

graphics_parameters G.26
identification_select G.2
indentured_list_method G.4
item_identifying_number_select G.15
item_list_applies_to_select G.9
item_or_element G.4
list_entry_select G.5
quantity_accuracy_select G.21
quantity_type_select G.9
revision_authorization_select G.19
shape_select G.16
source_identification G.7
usage_condition G.2
usage_context G.16
valid_tdp_elements_for_data_list G.11

ENTITY Figure
accessed_file G.6
alternate_element_identification G.20
alternate_identification_of_item G.20
alternate_item G.16
annotation G.29
approval G.23
assembly_relationship G.10
associated_list G.1
body_presentation G.27
certification G.23
change_identification G.19
column_header G.27
columnar_data_content_holder G.27
company G.25
company_code G.21
configuration G.18
content_property G.24
contract G.18
contract_submission G.7
data_definition_entry_item G.3
data_definition_entry_tdp_element G.4
data_definition_exchange G.1
data_definition_exchange_body G.3
data_definition_exchange_header G.2
data_definition_exchange_simple_entry G.3
data_definition_indentured_entry G.3
data_definition_indentured_list_method G.3
data_list G.11
data_list_body G.11
data_list_entry G.11
data_list_header G.11
data_list_tabulation G.11

date G.33
date_effectivity G.22
delivery_accounting G.2
design_authority G.17
distribution_notice G.18
document_list G.1
document_usage_parameter G.31
drawing G.29
drawing_suffix_number_combination G.15
effectivity G.22
element_identification G.17
element_type G.21
event G.25
exchange_file G.6
explicit_graphics G.26
external_graphics_file G.26
external_library_reference G.30
field_content_holder G.28
field_content_holder_singular G.28
field_content_holder_tabulation G.28
field_location G.28
field_location_definition G.28
field_title G.28
file G.6
file_format G.6
file_relationship G.7
foreign_defined_item G.9
general_item_definition_relationship G.10
geometric_validation_property G.30
geometry G.29
group_parent_column_header G.27
group_parent_content_holder G.27
group_parent_header G.27
header G.17
header_block G.27
header_configuration_with_element_identification G.17
header_presentation G.27
indentured_data_list G.12
indentured_data_list_body G.12
indentured_data_list_entry G.12
indentured_data_list_header G.12
indentured_data_list_tabulation G.12
indentured_list_by_document G.4
indentured_list_by_part G.4
indentured_list_by_part_with_document_references_to_parts G.4
independent_property G.30
independent_property_relationship G.30
independent_property_usage G.30
index_list G.13

index_list_body G.13
index_list_entry G.13
index_list_header G.13
item G.16
item_identification G.15
item_list G.9
item_list_drawing G.9
item_parent_to_item_child_relationship G.5
item_parent_to_tdp_element_child_relationship G.5
item_type G.21
item_usage G.9
list_item_usage G.7
list_presentation G.26
load_point G.28
locally_defined_item G.9
location_instance G.10
lot_effectivity G.22
made_from_stock_material G.9
make_from_relationship G.10
notation G.24
number_with_units G.10
other_list G.14
other_list_body G.14
other_list_body_table G.14
other_list_header G.14
other_reference_document G.31
page_parameters G.26
page_presentation G.26
parts_list G8
parts_list_body G.8
parts_list_header G.8
part_occurrence_in_assembly G.10
person G.25
person_and_organization G.25
product_configuration G.22
product_data_set G.29
product_data_set_with_format G.29
product_data_set_with_shading G.29
product_data_set_without_format G.29
product_model G.22
product_presentation G.29
project G.25
promissory_usage G.10
property_assignment G.30
quantified_part_usage_in_assembly G.10
quantity G.9
reason G.2
reference_document G.31
reference_document_usage G.31

relative_event G.25
release_authentication G.18
retrofit_usage G.8
revision G.8
security_classification G.18
sequence_effectivity G.22
shaded_shape_model G.29
shape_model G.29
sheet G.29
simple_list_of_elements G.3
simple_list_of_files G.3
single_document_list G.1
size characteristics G.7
size_characteristics_full_size G.7
size_characteristics_internal_divisions G.7
size_characteristics_sheet_across_file G.7
source_file G.6
source_information_type G.24
source_location G.7
special_condition G.24
specification_document G.31
specified_part_in_assembly_tree G.10
standard_document G.31
status G.19
stock_material G.9
stock_size G.10
stock_size_parameters G.10
string_location_definition G.28
system_declaration G.6
system_destination G.2
tabulation G.27
tabulation_entry G.14
tabulation_header G.27
tdp_element G.1
tdp_element_and_item_association G.1
tdp_element_list_item G.8
tdp_element_parent_to_item_child_relationship G.5
tdp_element_parent_to_tdp_element_child_relationship G.5
tdp_indentured_item G.5
tdp_indentured_tdp_element G.5
time G.25
time_interval_effectivity G.22
transformation G.10
volume G.9
weight G.16

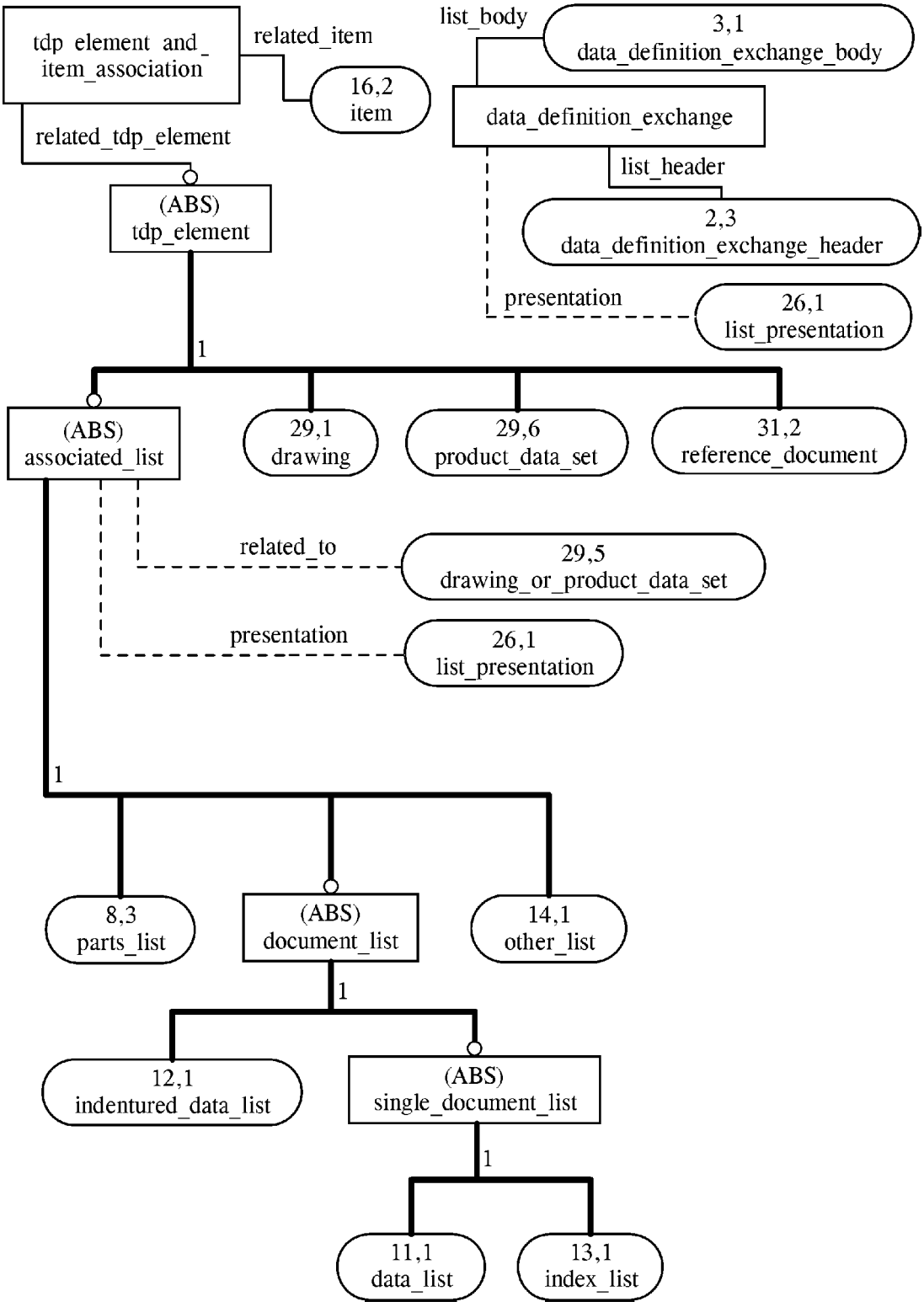

Figure G.1 — ARM diagram (1 of 33)

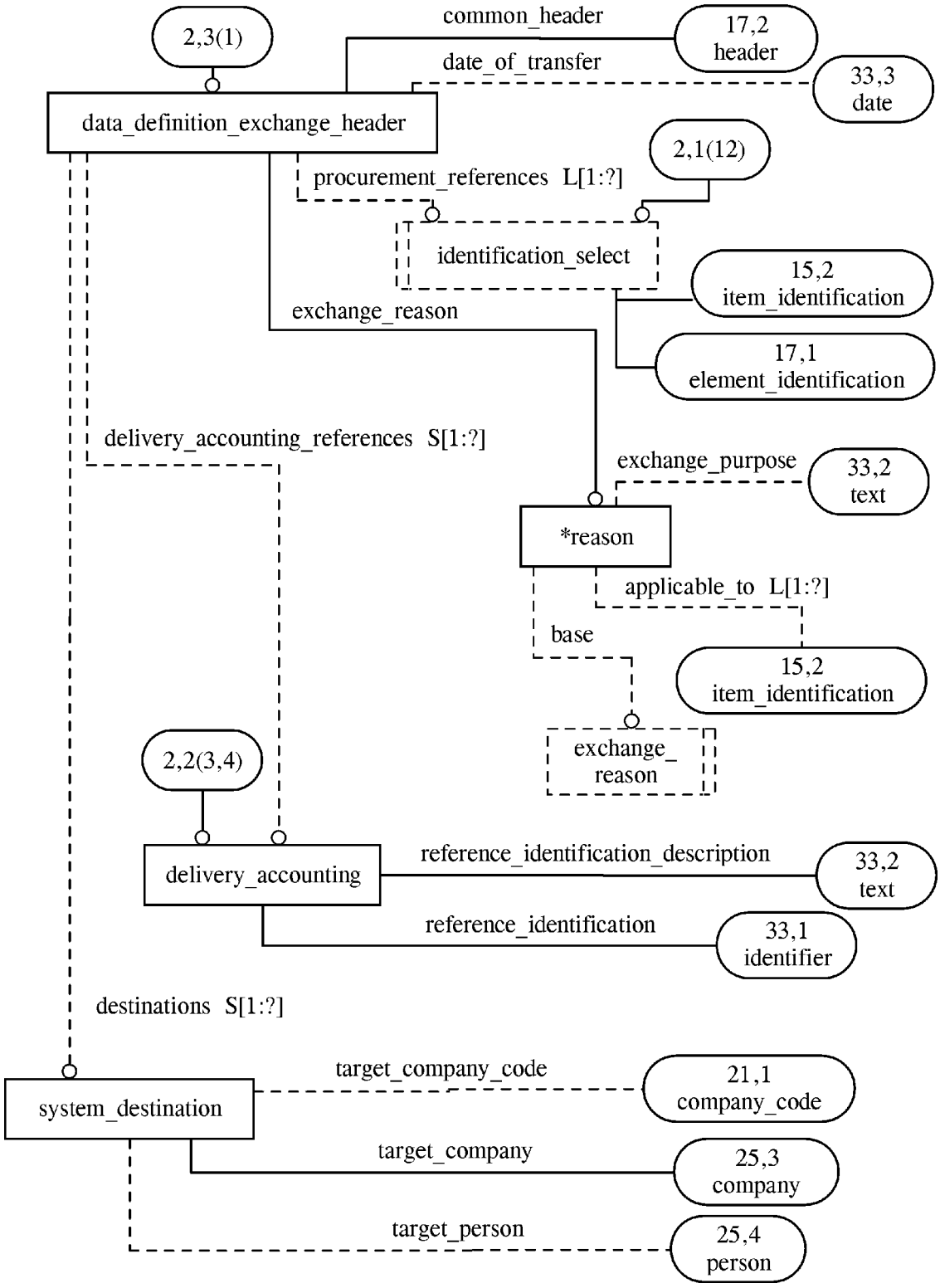

Figure G.2 — ARM diagram (2 of 33)

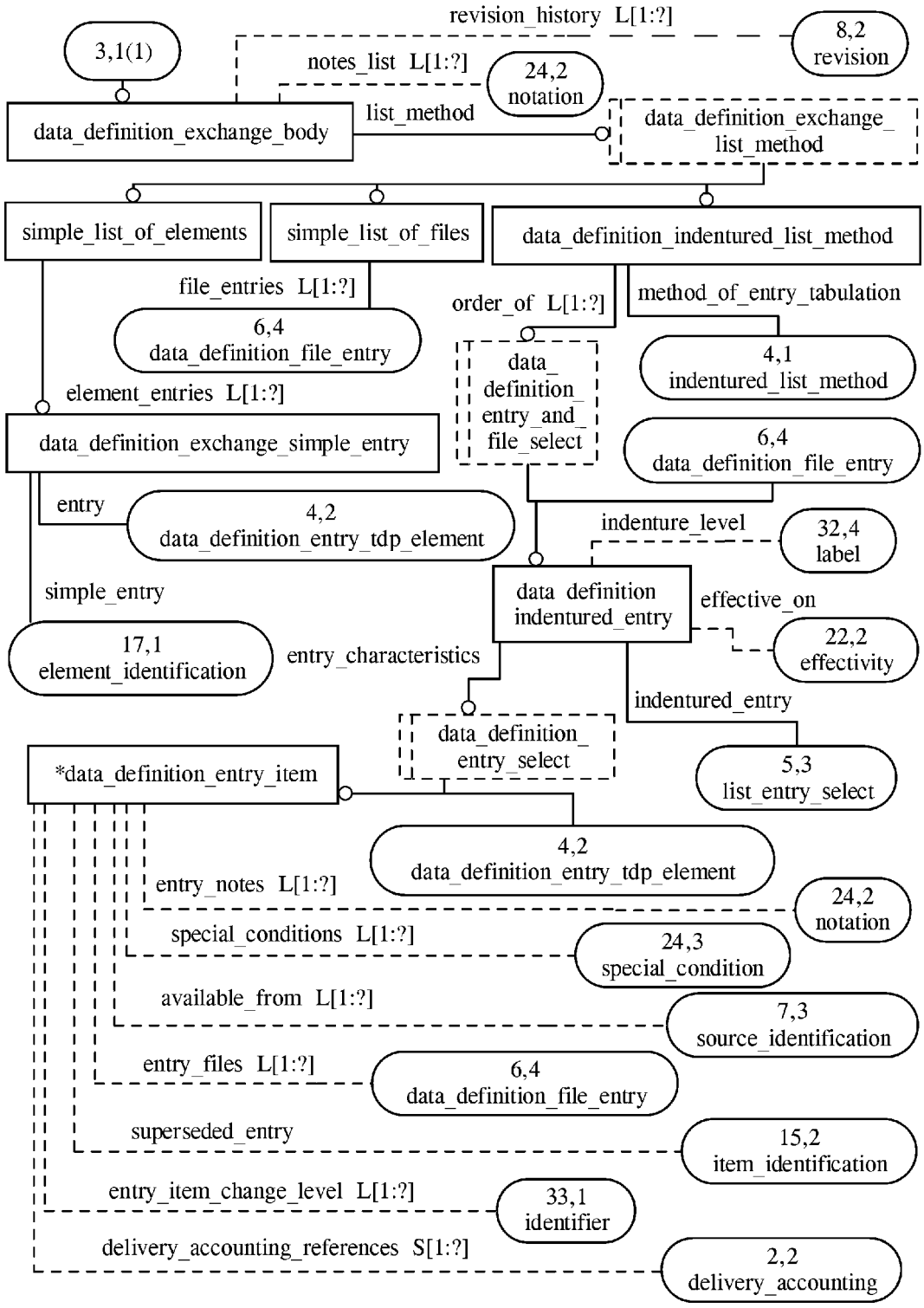

Figure G.3 — ARM diagram (3 of 33)

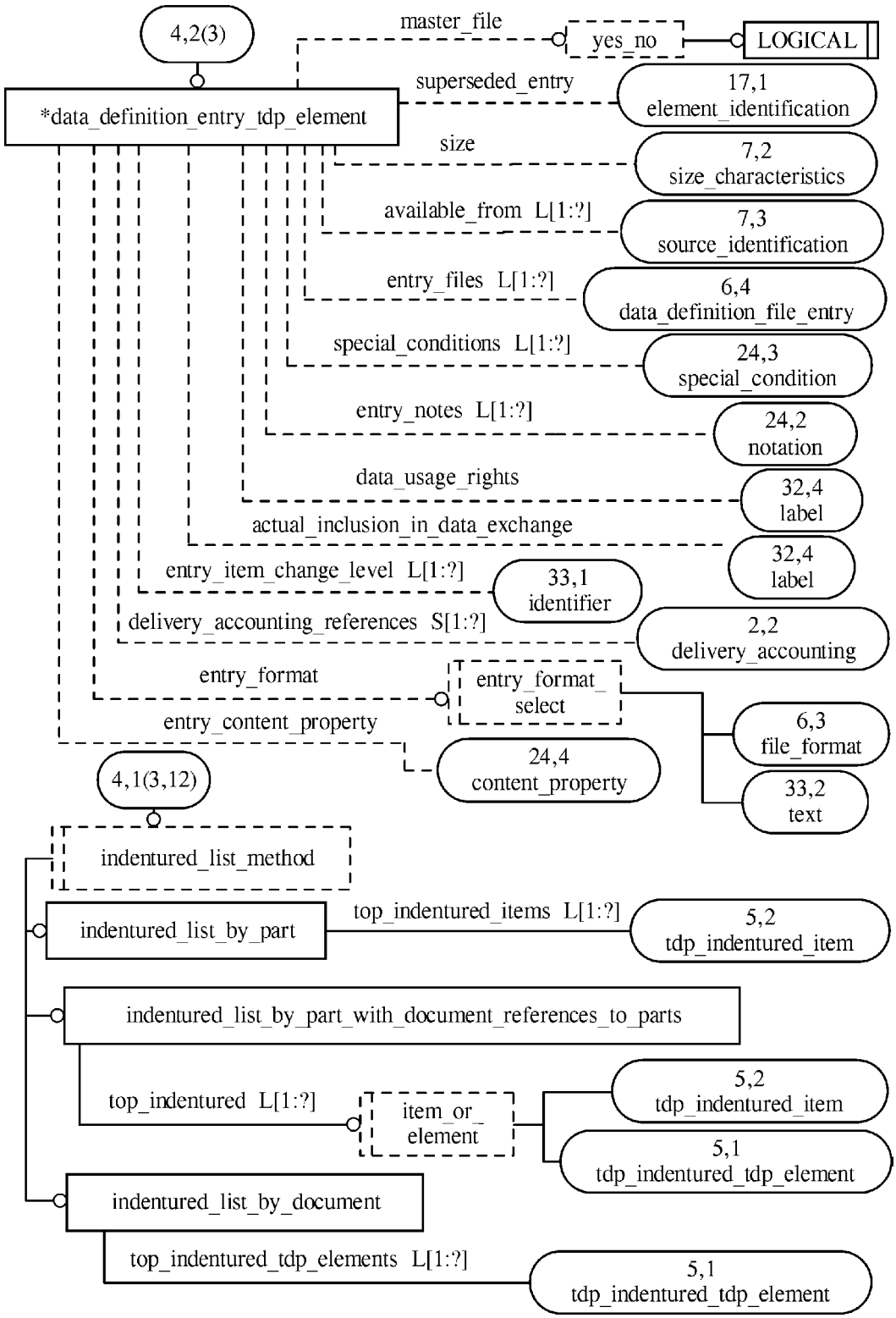

Figure G.4 — ARM diagram (4 of 33)

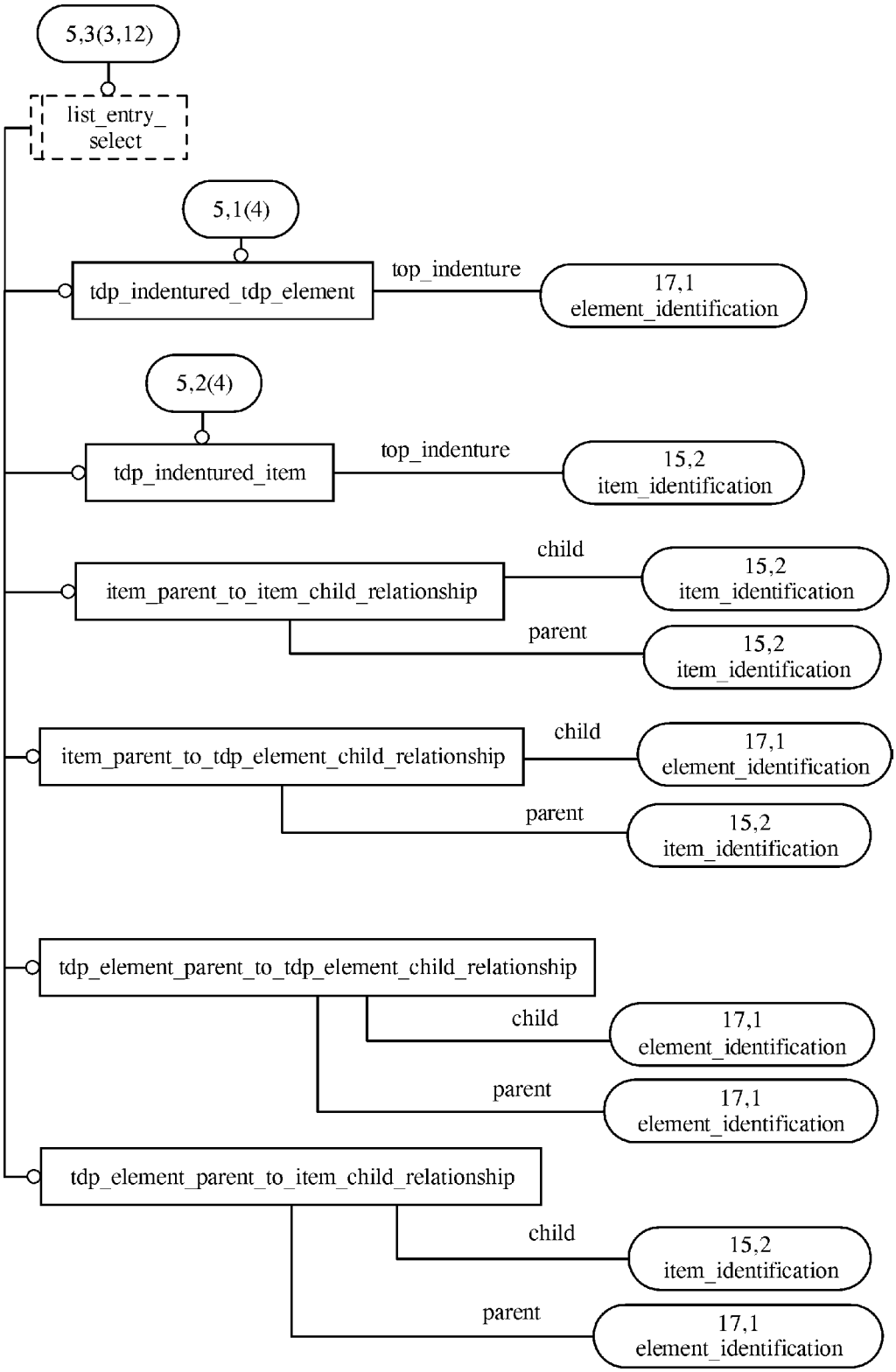

Figure G.5 — ARM diagram (5 of 33)

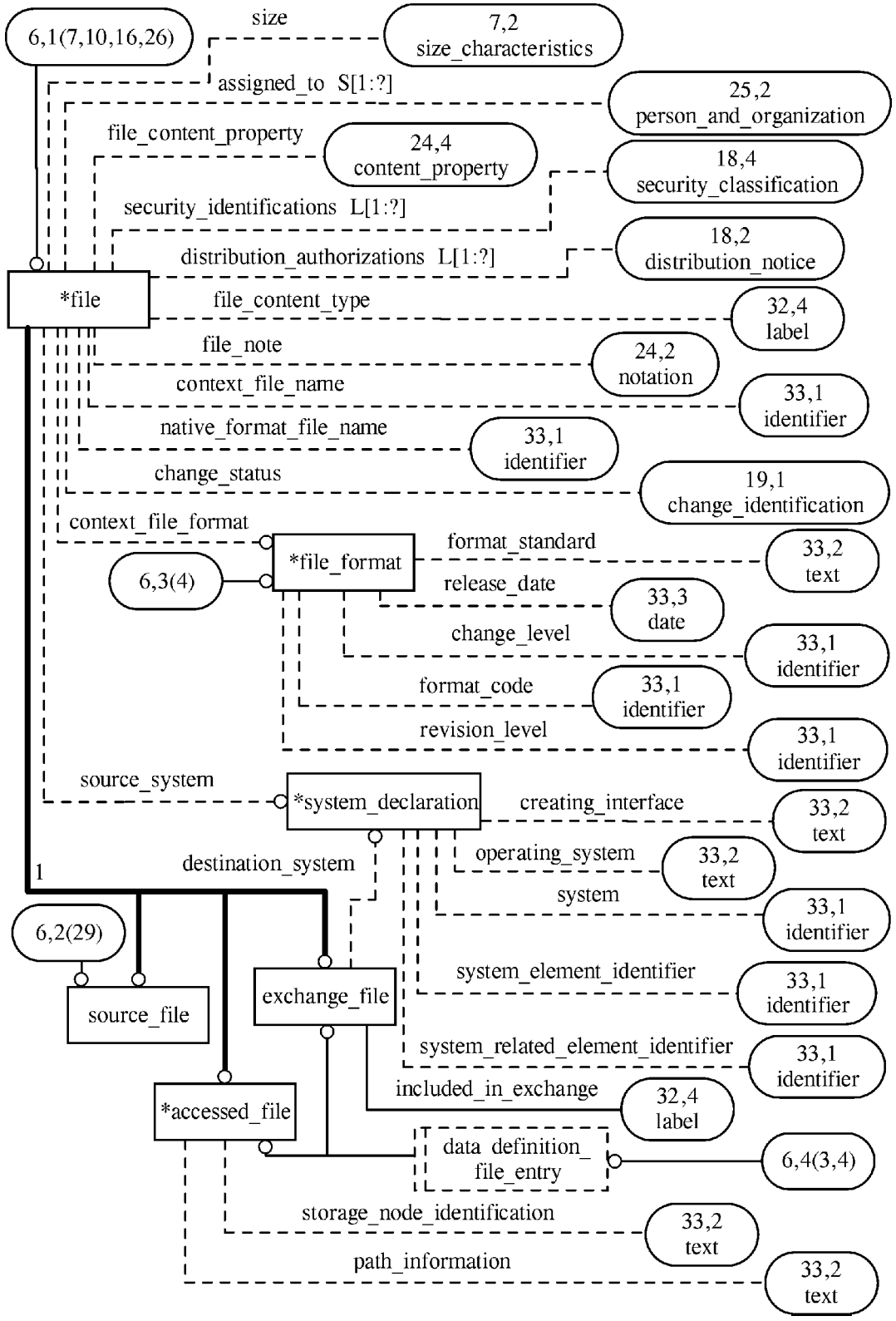

Figure G.6 — ARM diagram (6 of 33)

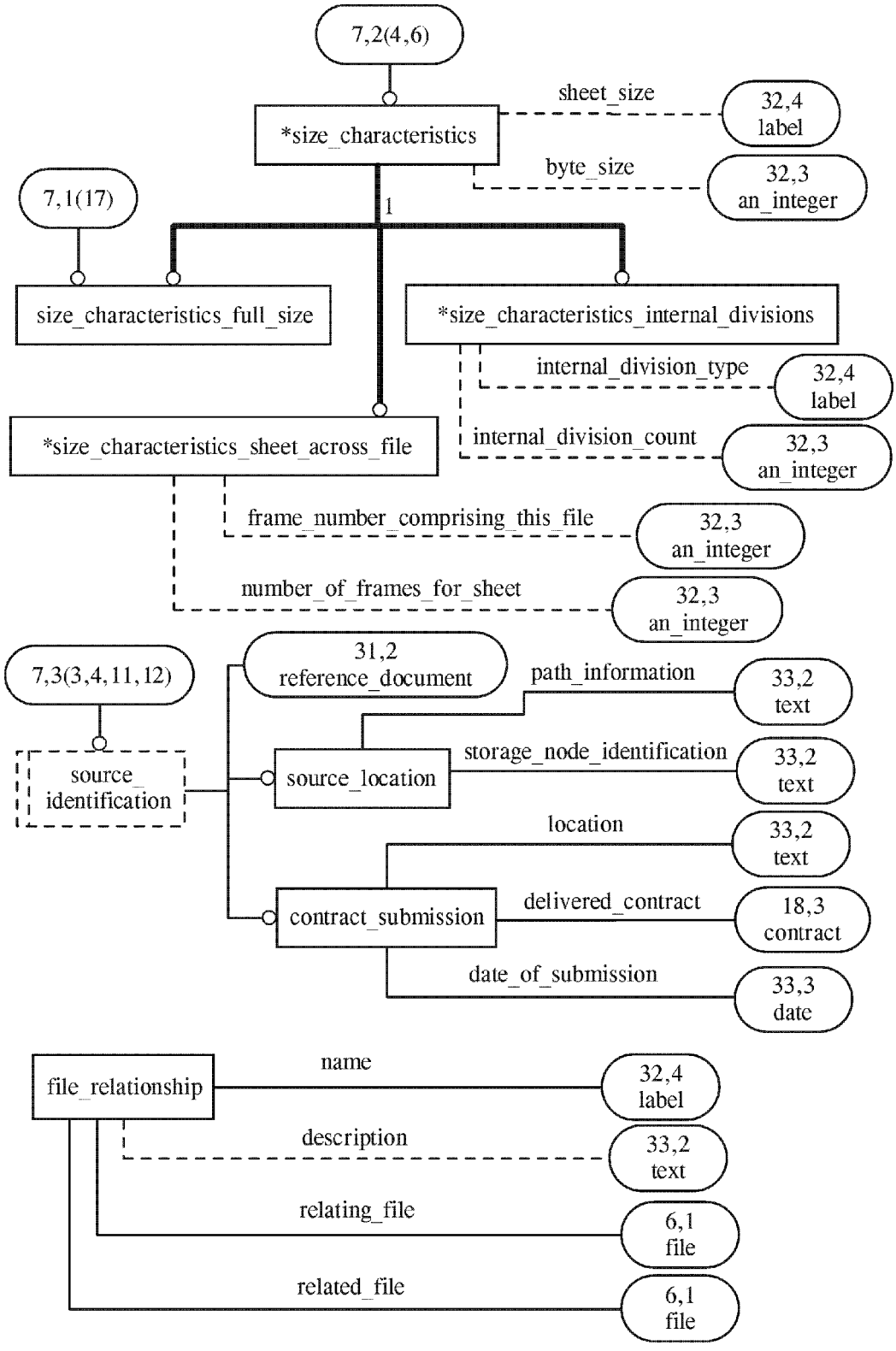

Figure G.7 — ARM diagram (7 of 33)

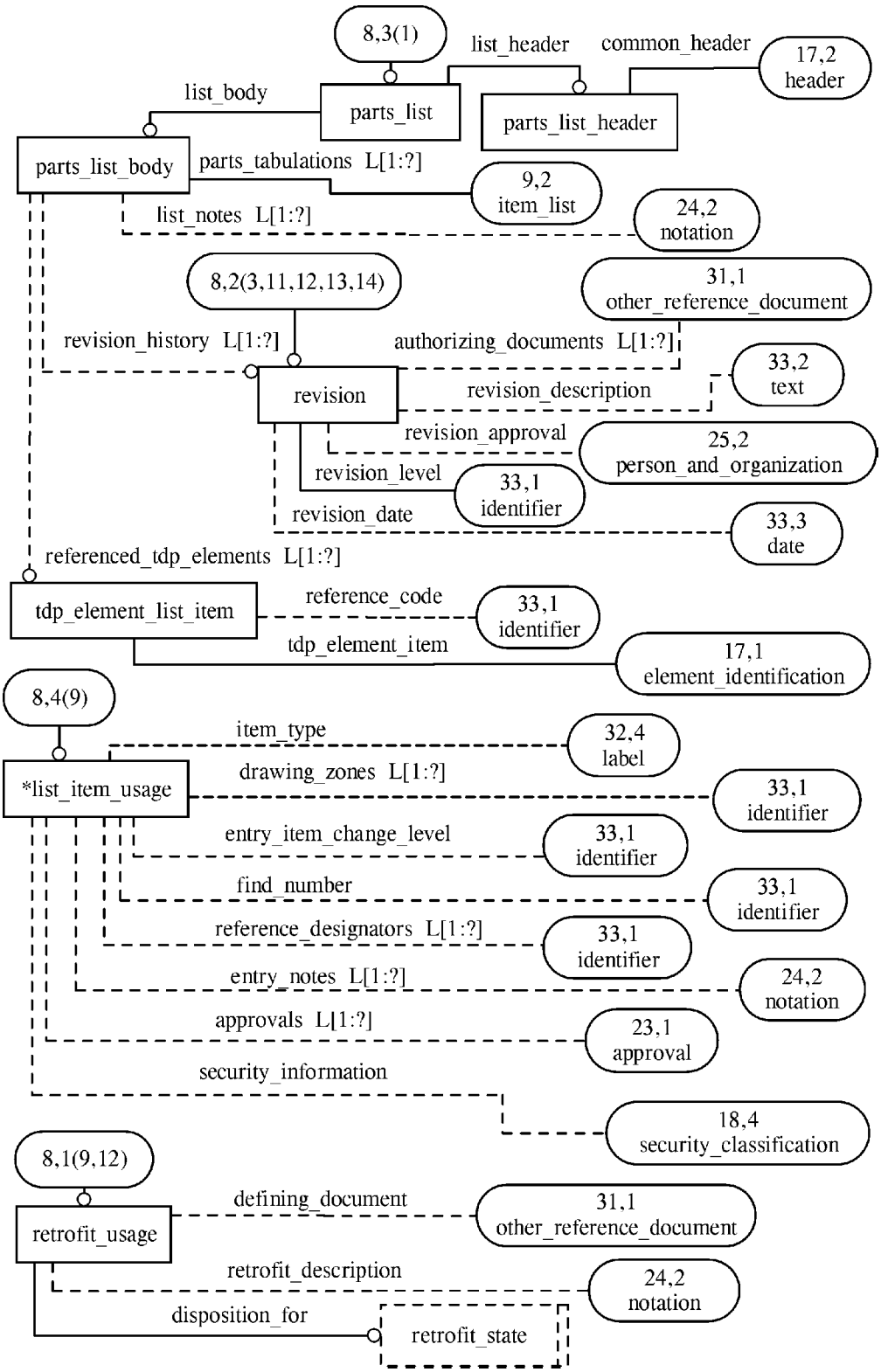

Figure G.8 — ARM diagram (8 of 33)

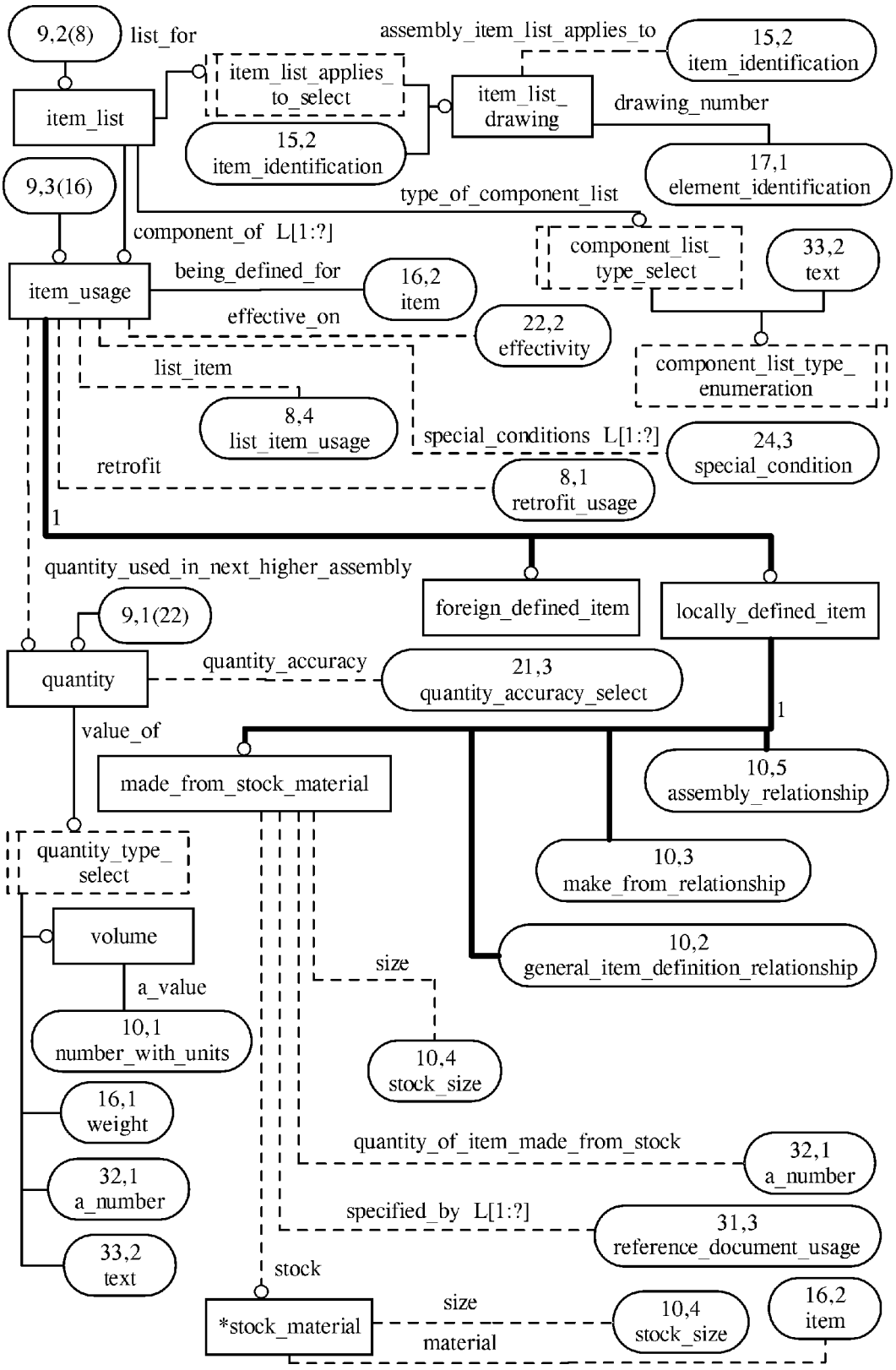

Figure G.9 — ARM diagram (9 of 33)

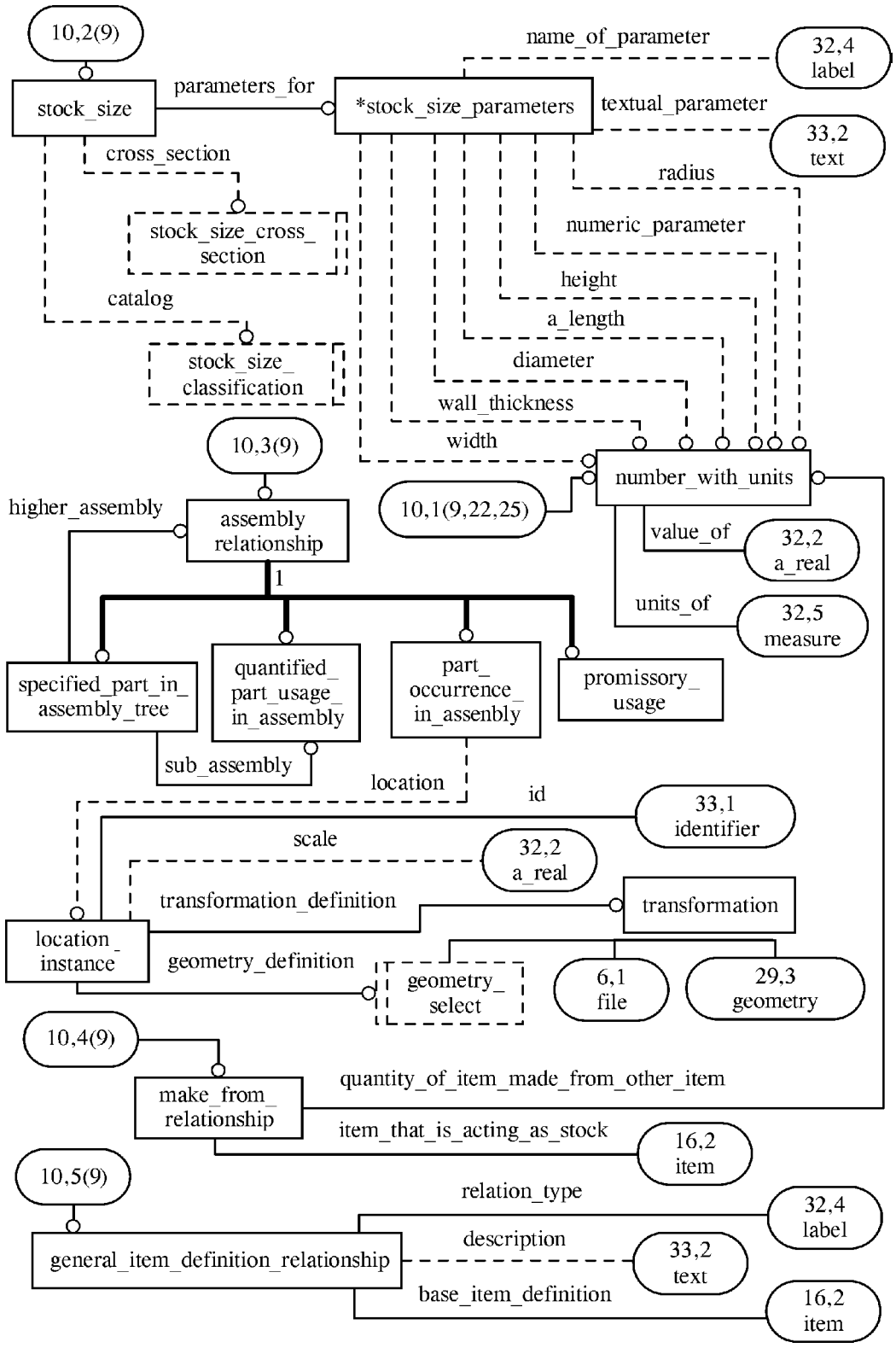

Figure G.10 — ARM diagram (10 of 33)

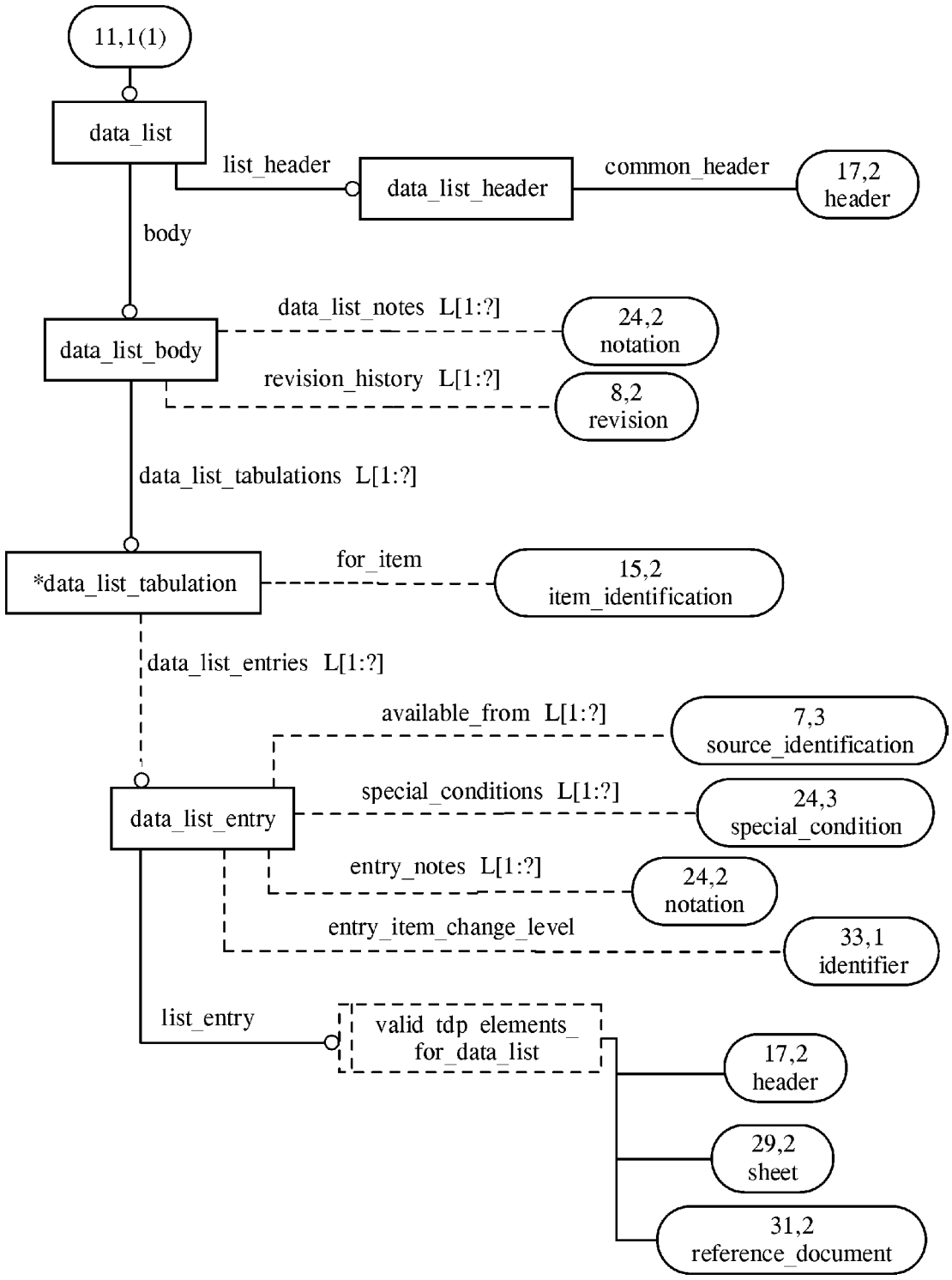

Figure G.11 — ARM diagram (11 of 33)

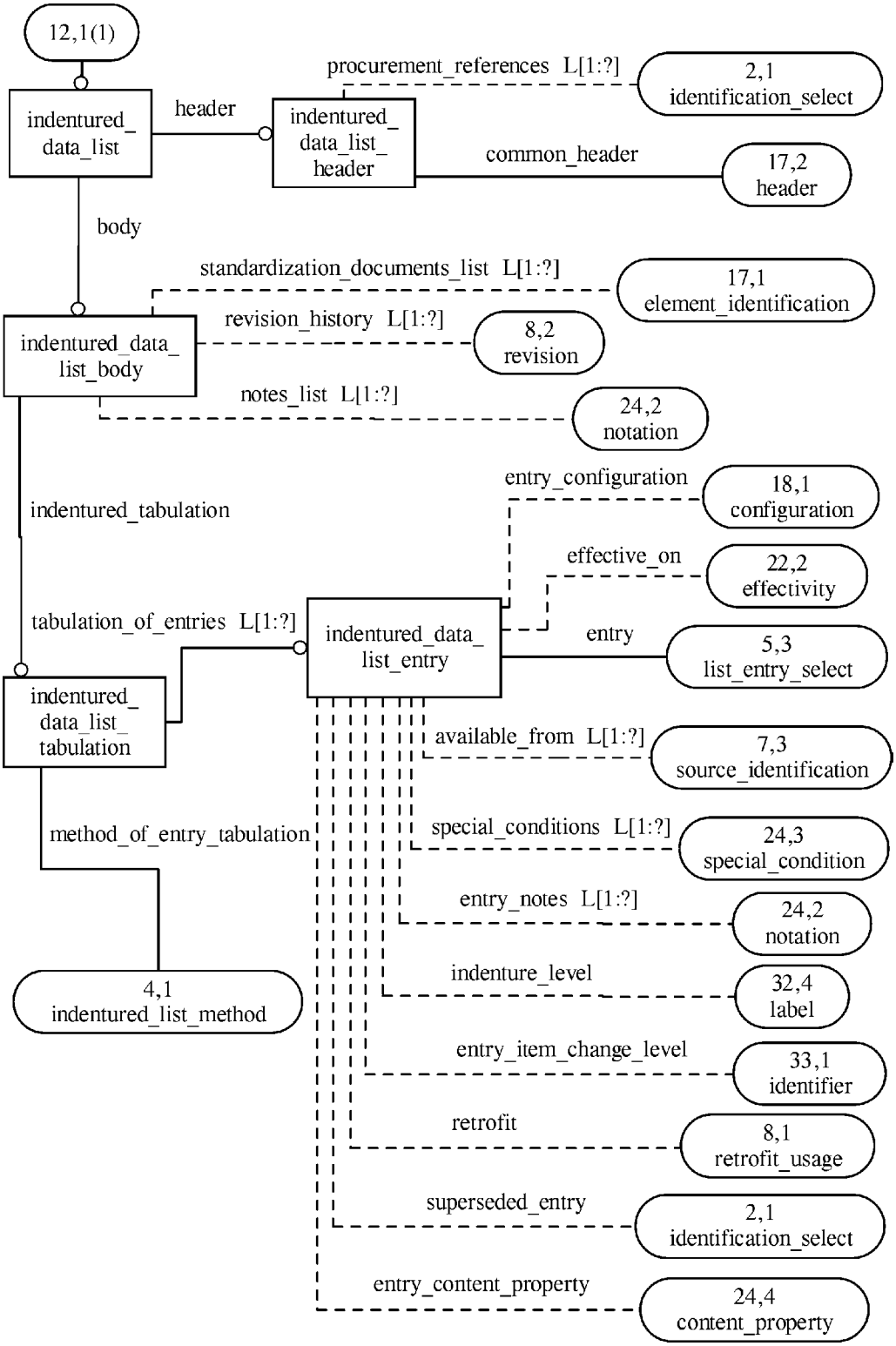

Figure G.12 — ARM diagram (12 of 33)

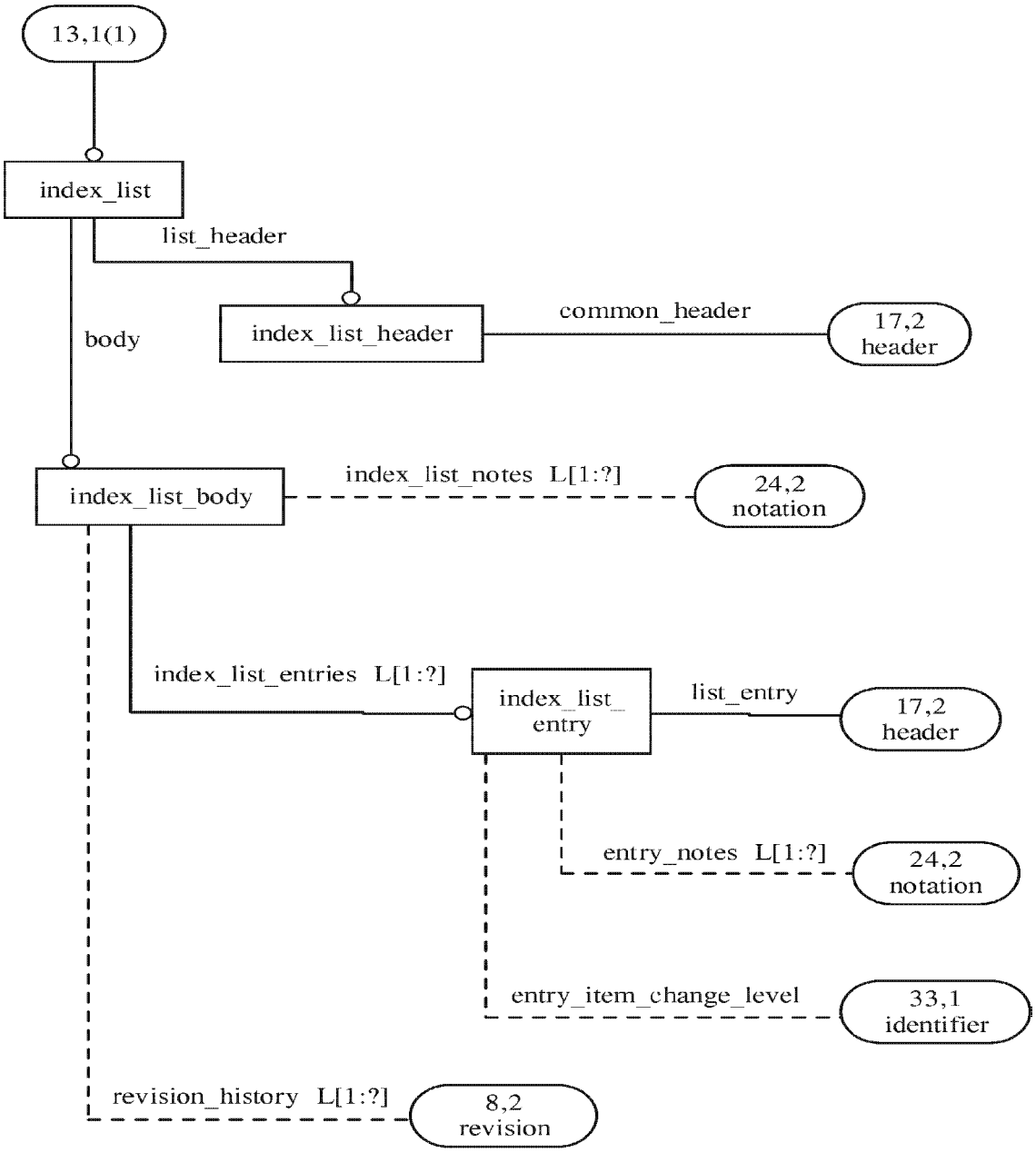

Figure G.13 — ARM diagram (13 of 33)

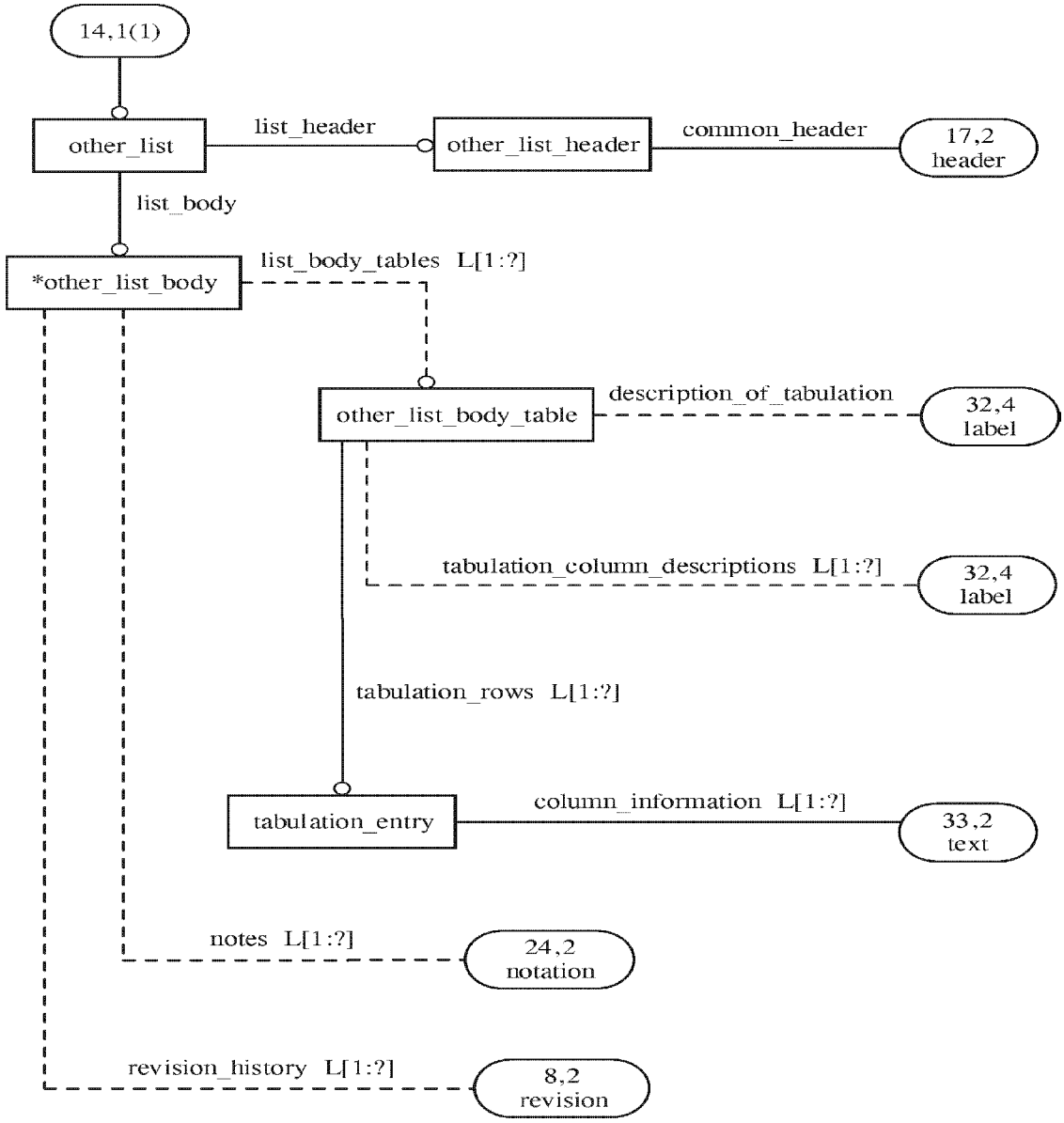

Figure G.14 — ARM diagram (14 of 33)

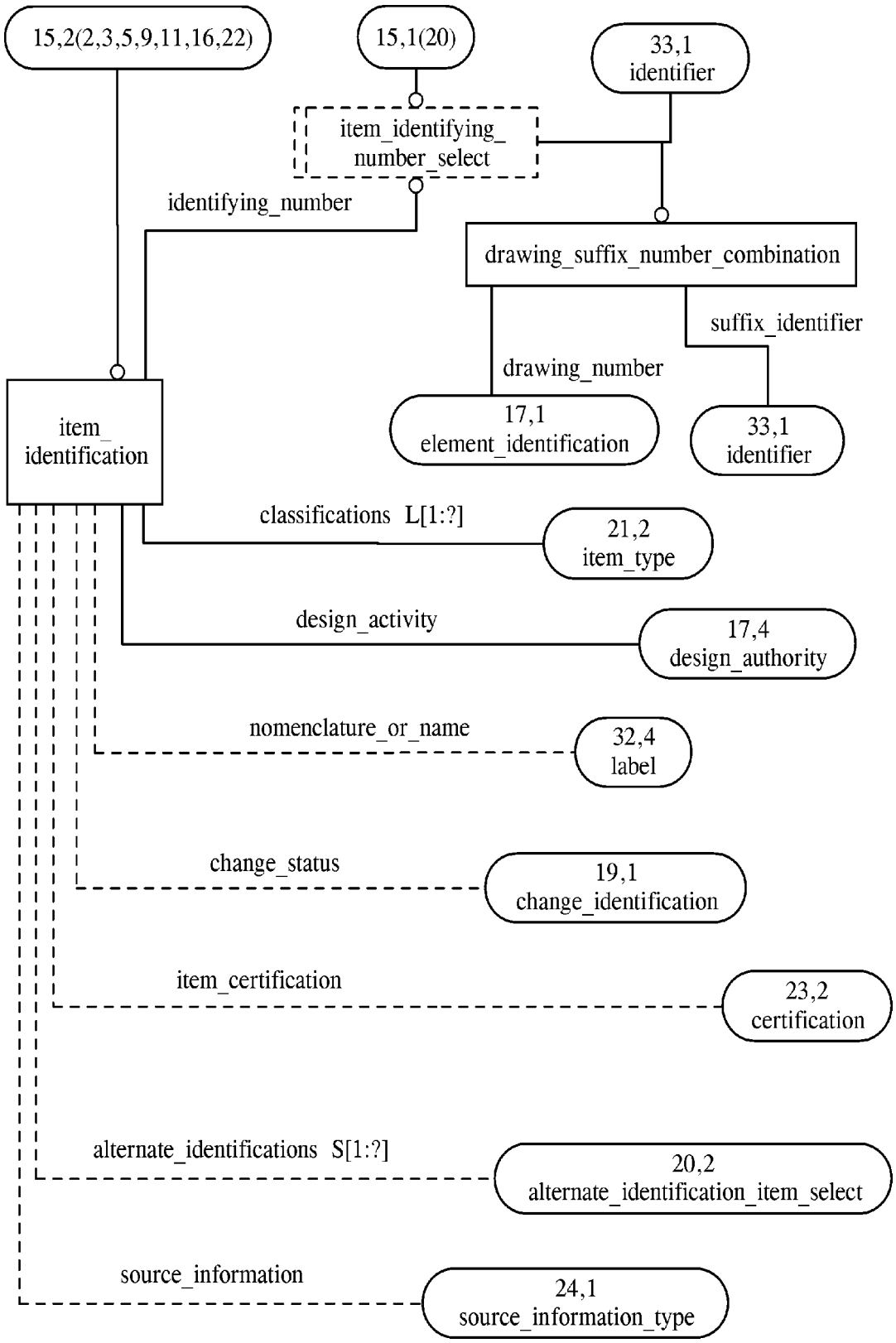

Figure G.15 — ARM diagram (15 of 33)

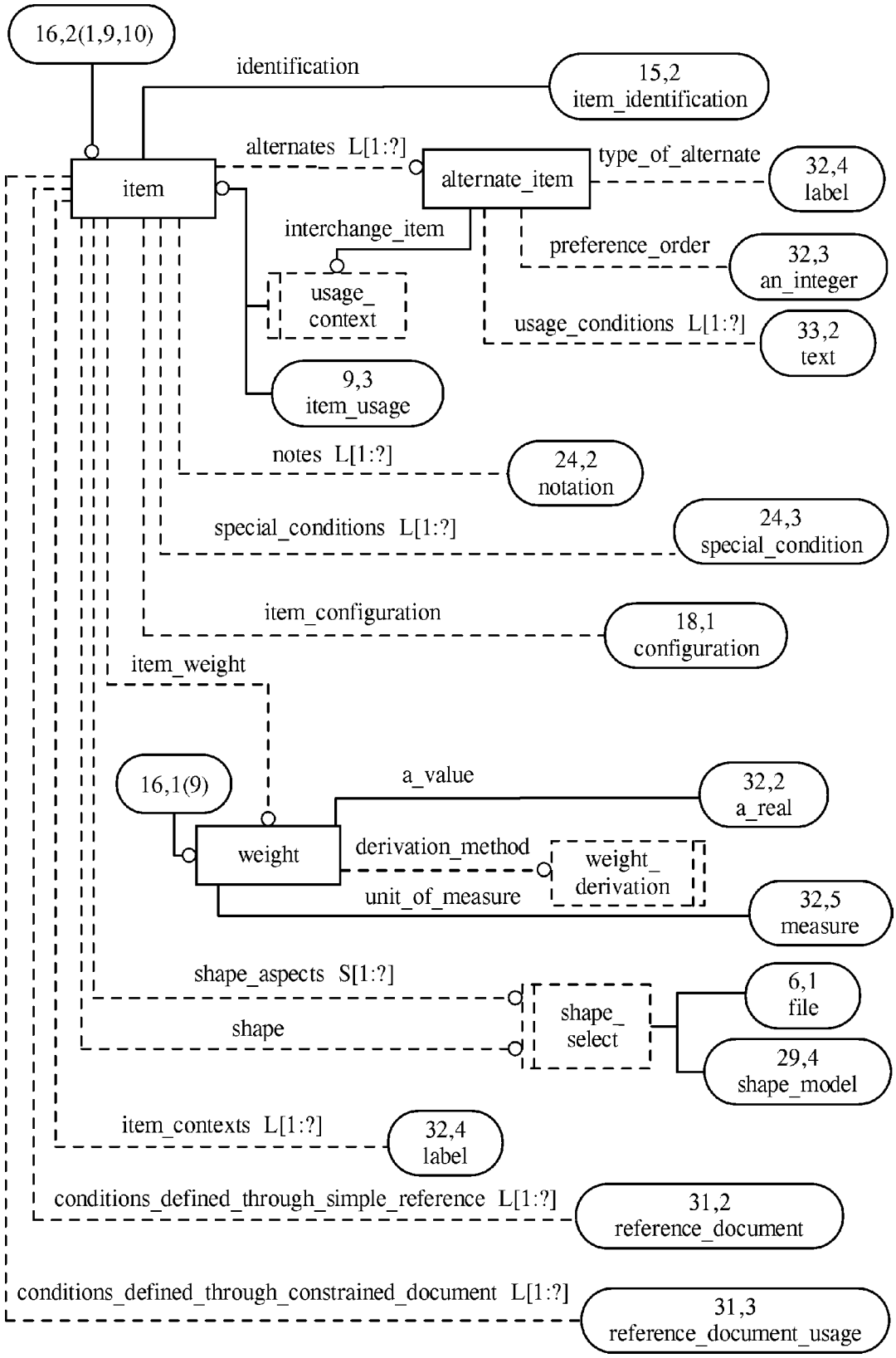

Figure G.16 — ARM diagram (16 of 33)

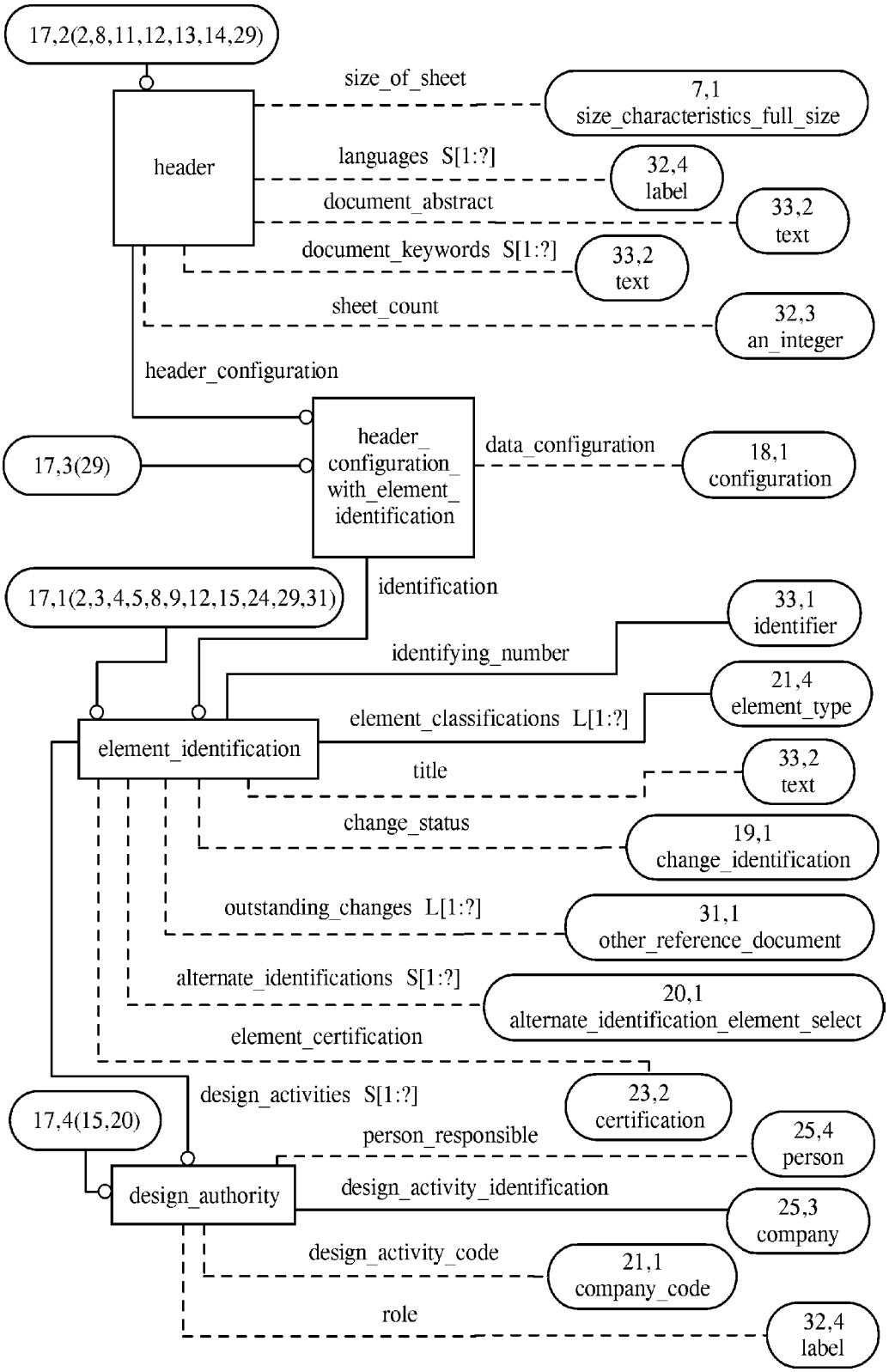

Figure G.17 — ARM diagram (17 of 33)

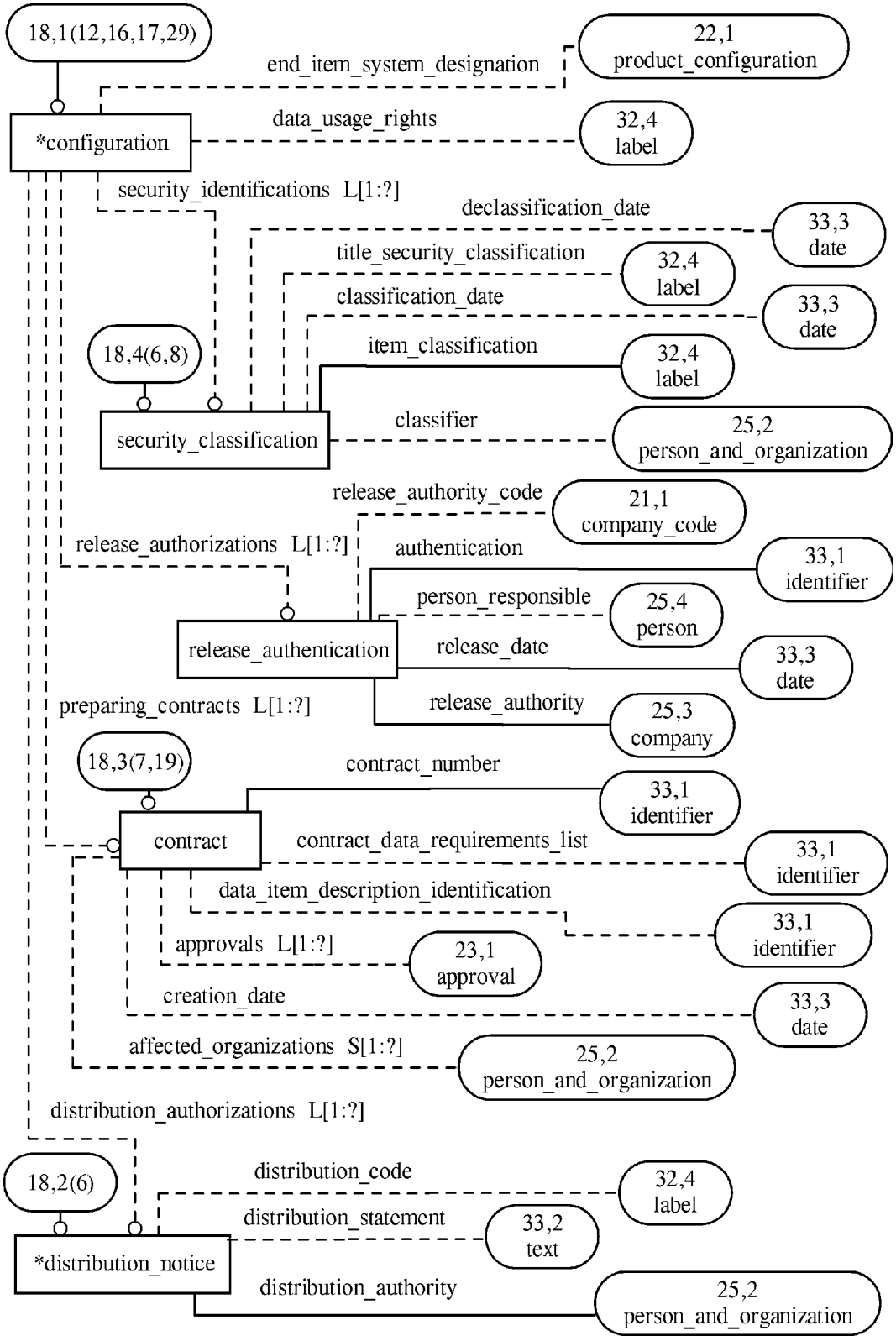

Figure G.18 — ARM diagram (18 of 33)

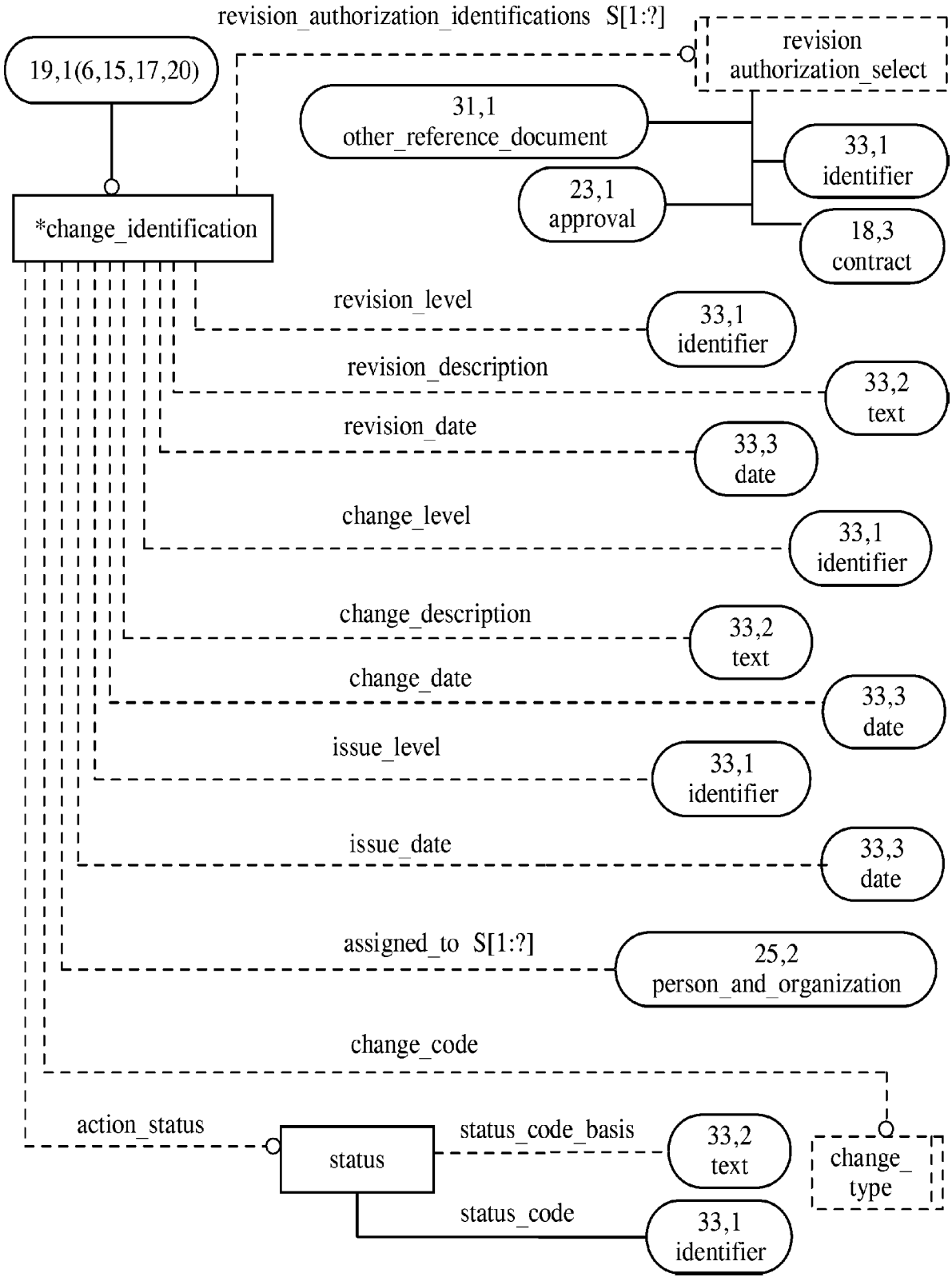

Figure G.19 — ARM diagram (19 of 33)

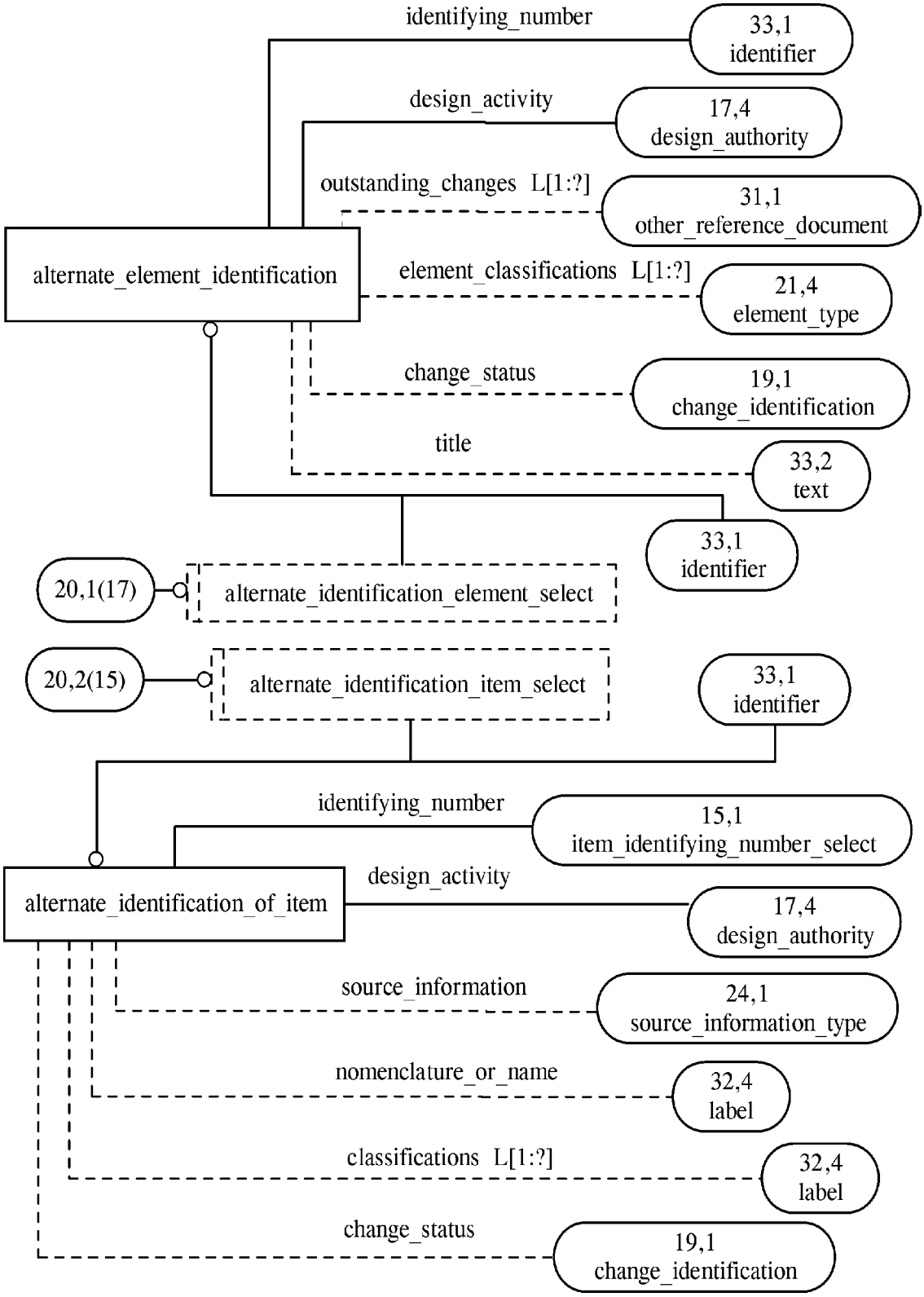

Figure G.20 — ARM diagram (20 of 33)

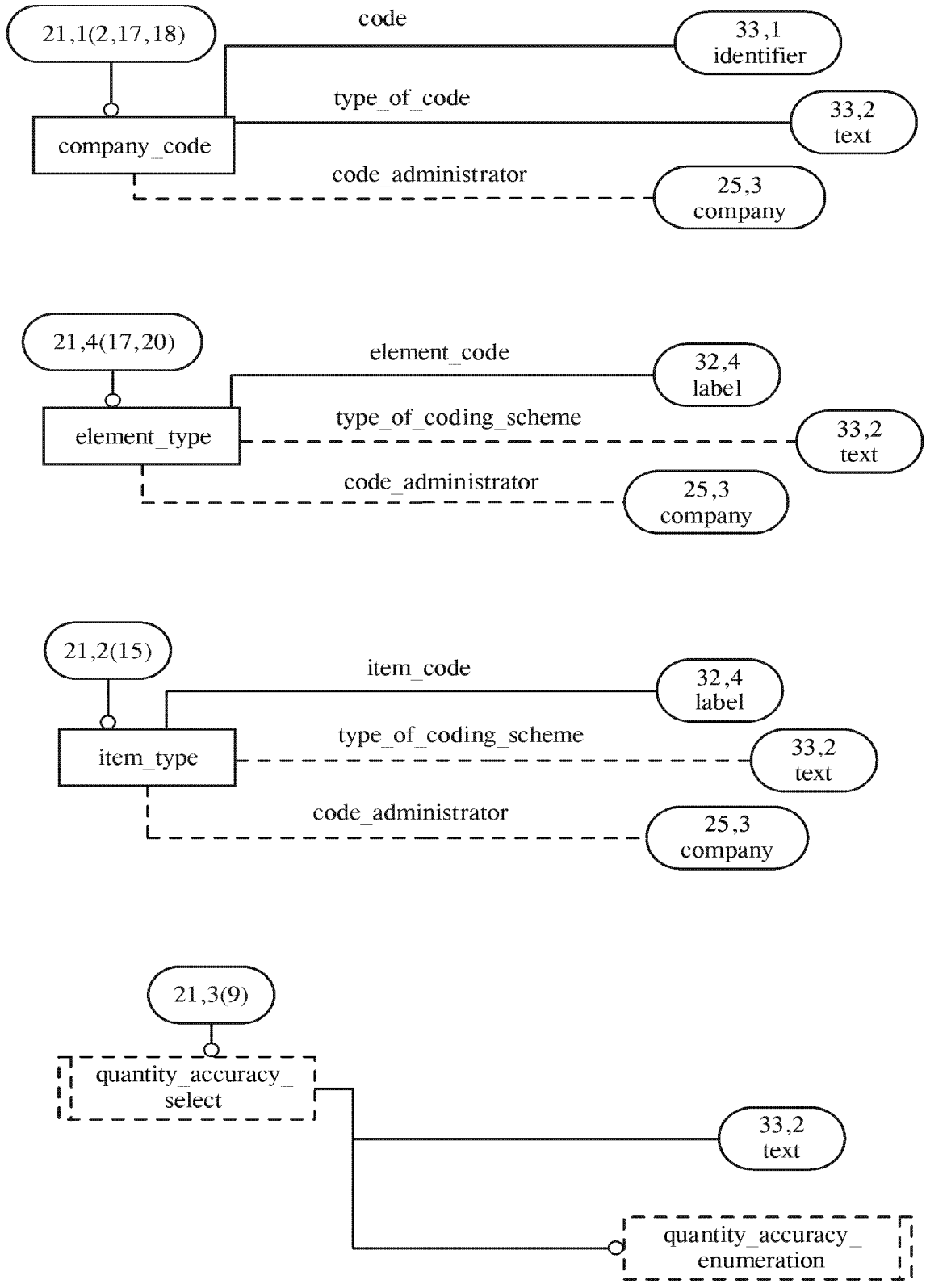

Figure G.21 — ARM diagram (21 of 33)

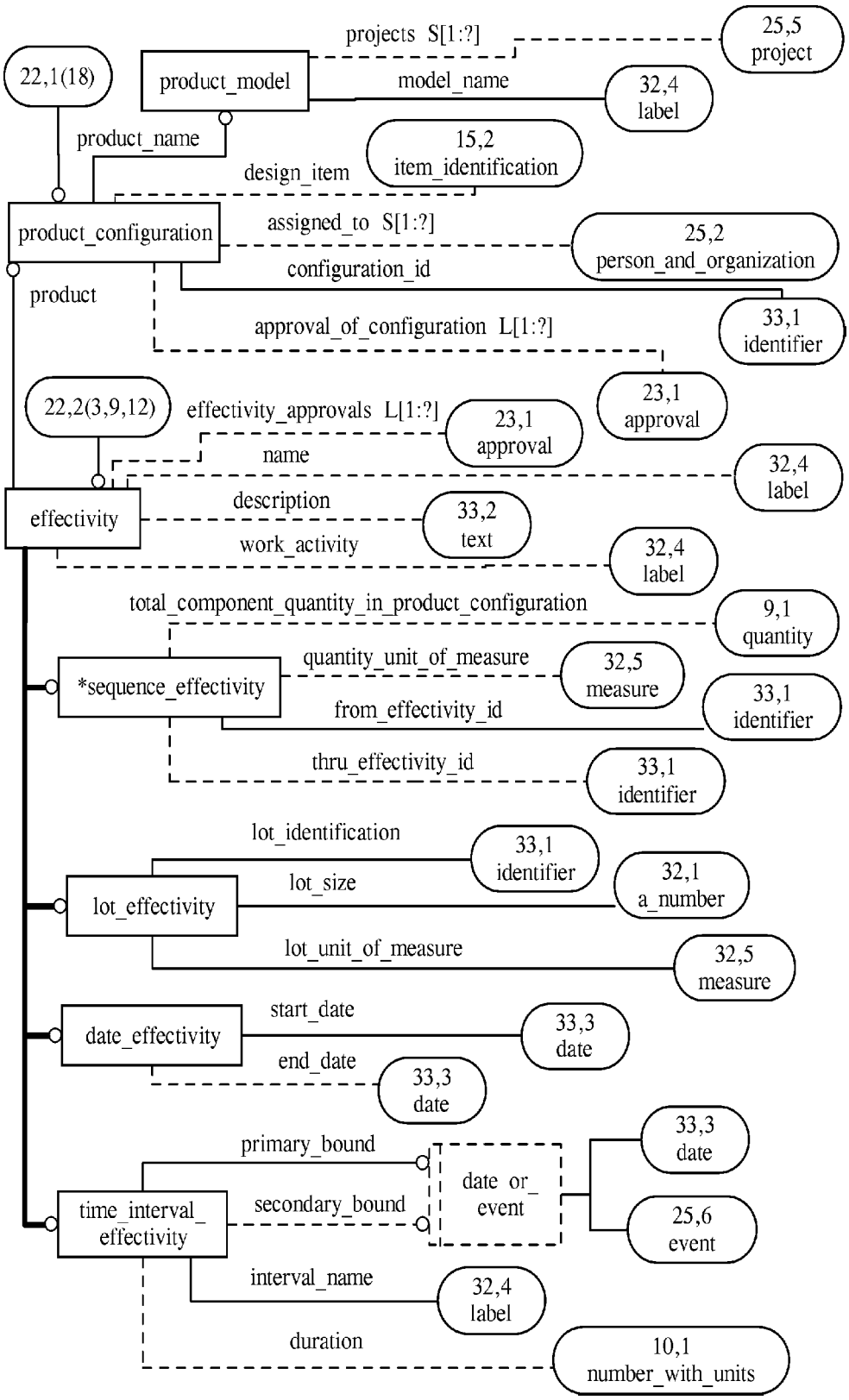

Figure G.22 — ARM diagram (22 of 33)

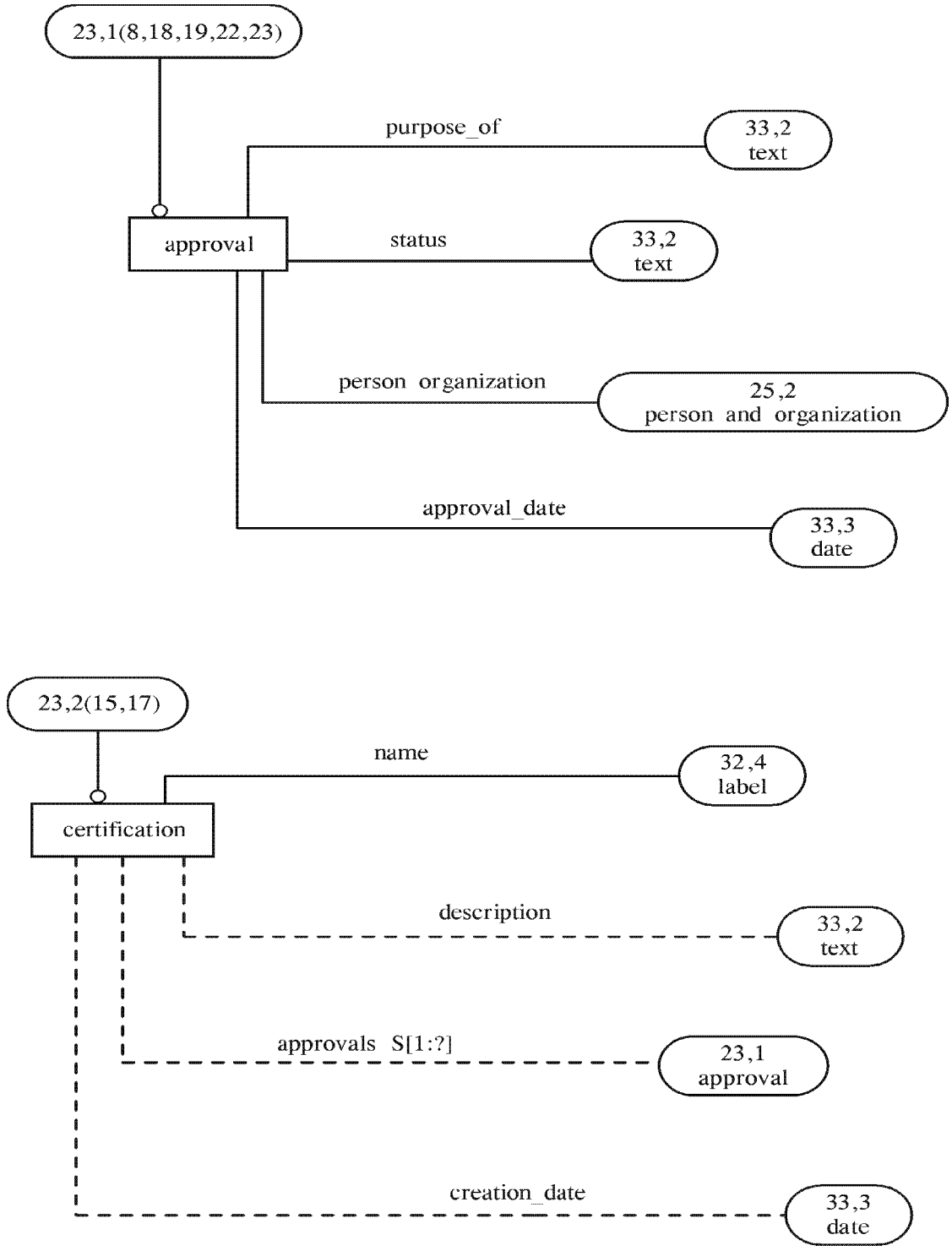

Figure G.23 — ARM diagram (23 of 33)

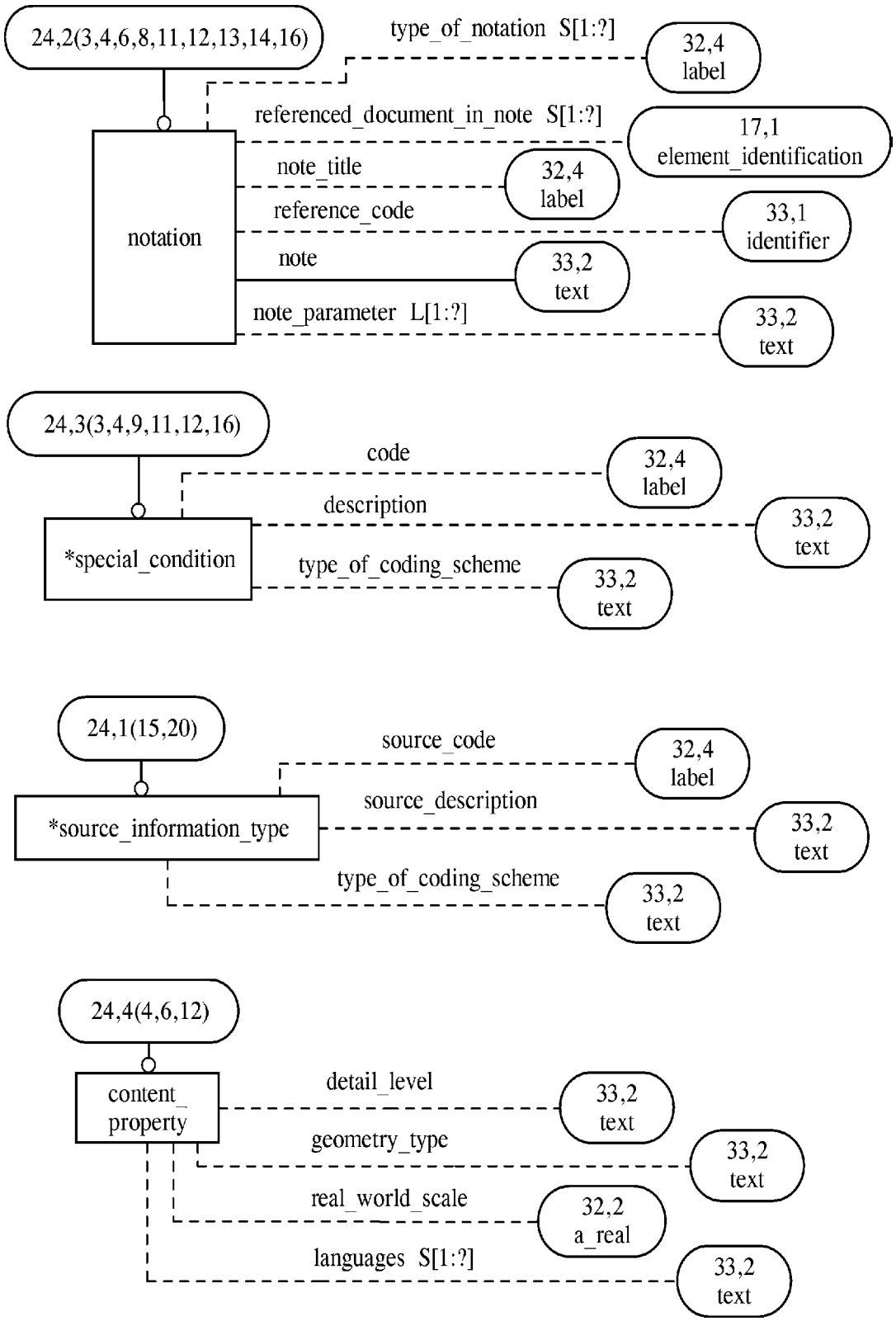

Figure G.24 — ARM diagram (24 of 33)

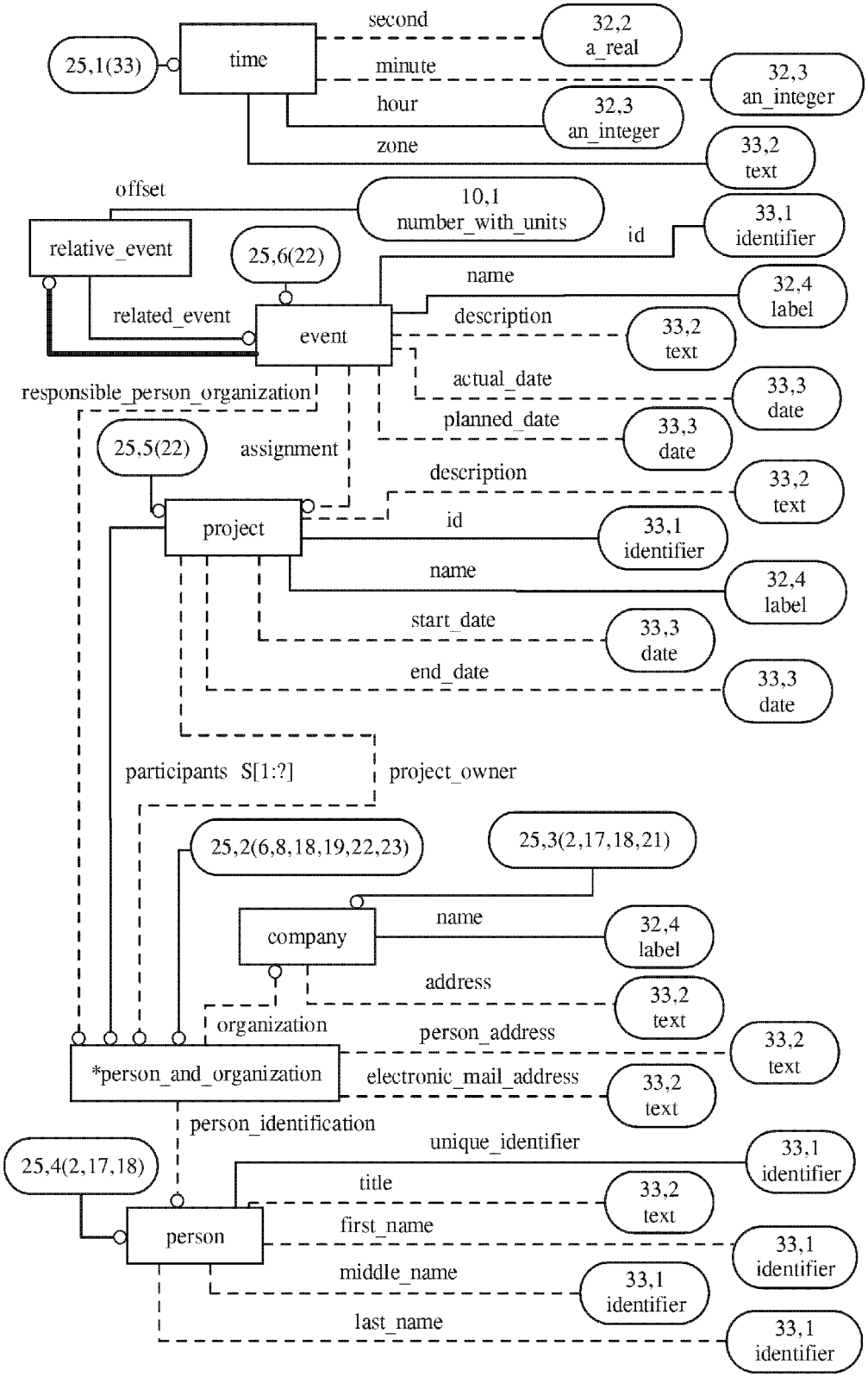

Figure G.25 — ARM diagram (25 of 33)

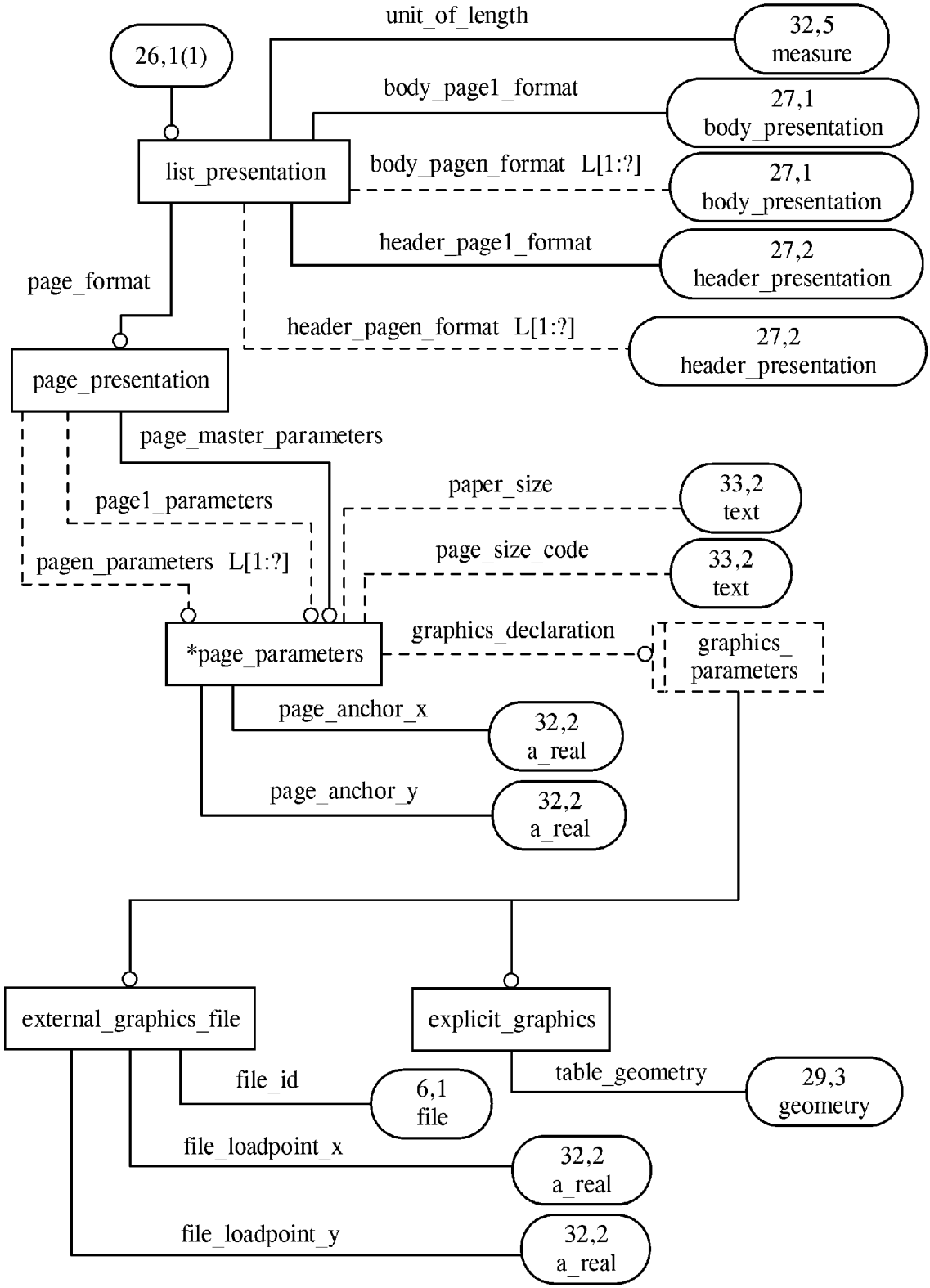

Figure G.26 — ARM diagram (26 of 33)

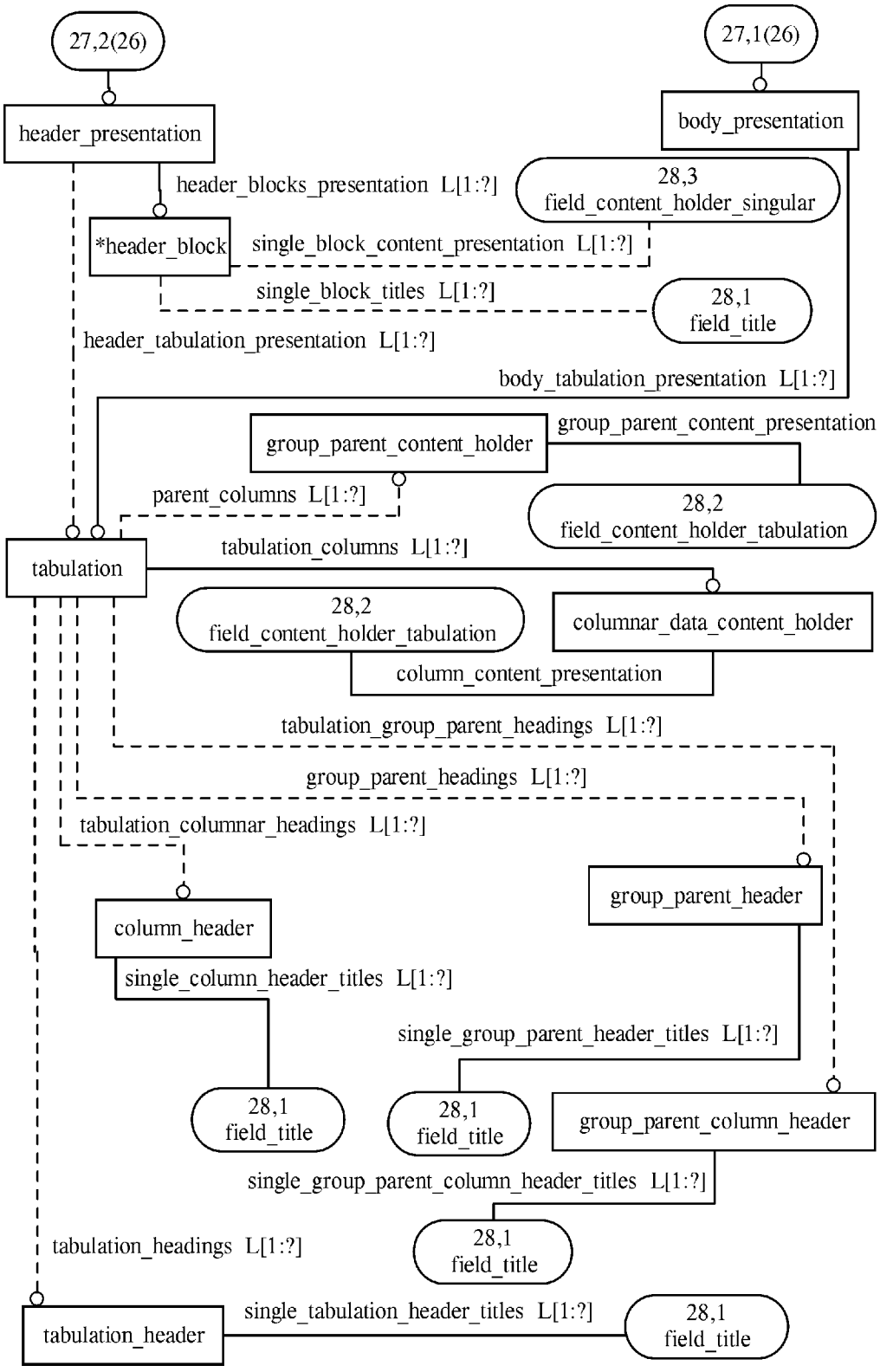

Figure G.27 — ARM diagram (27 of 33)

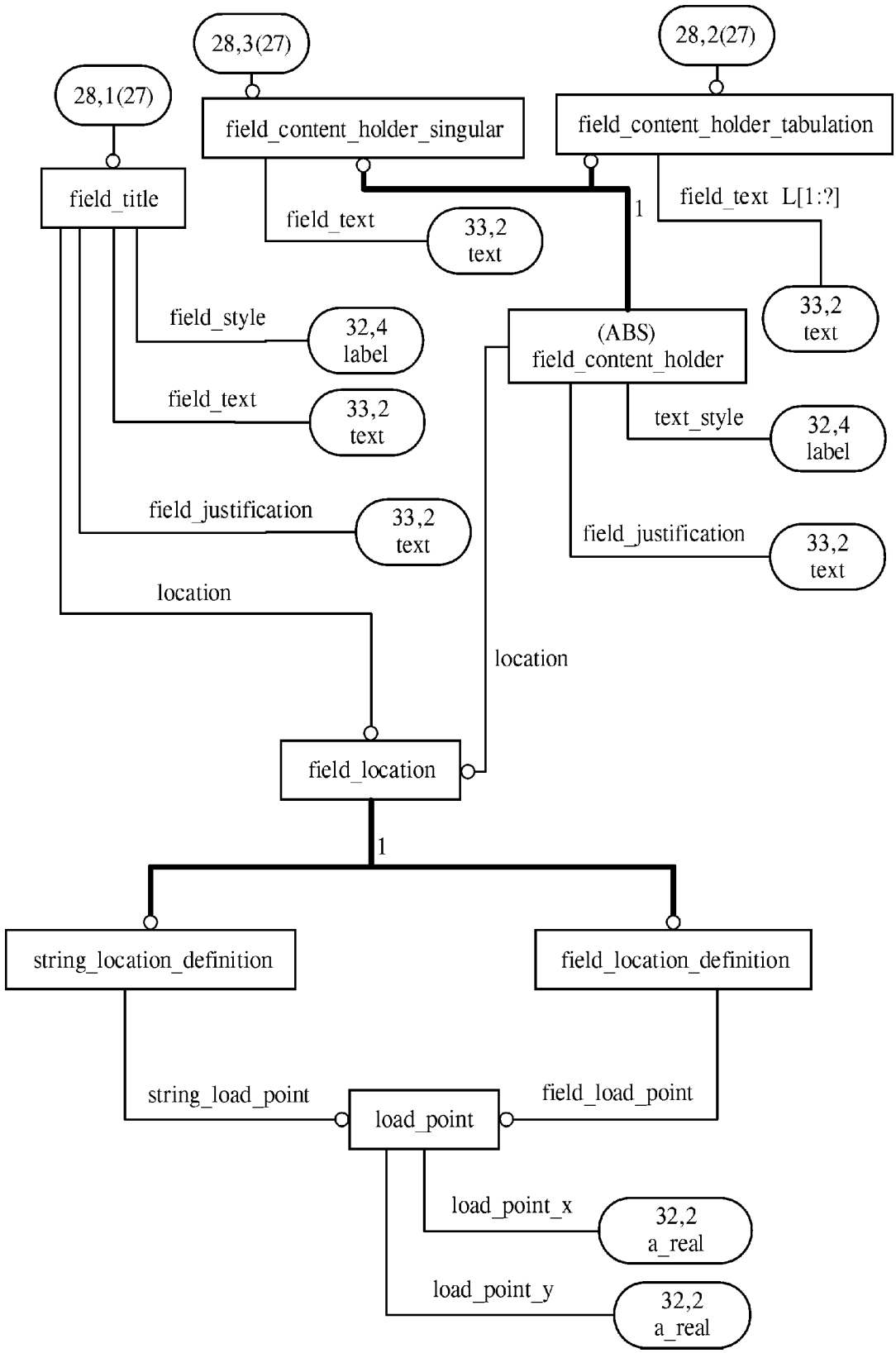

Figure G.28 — ARM diagram (28 of 33)

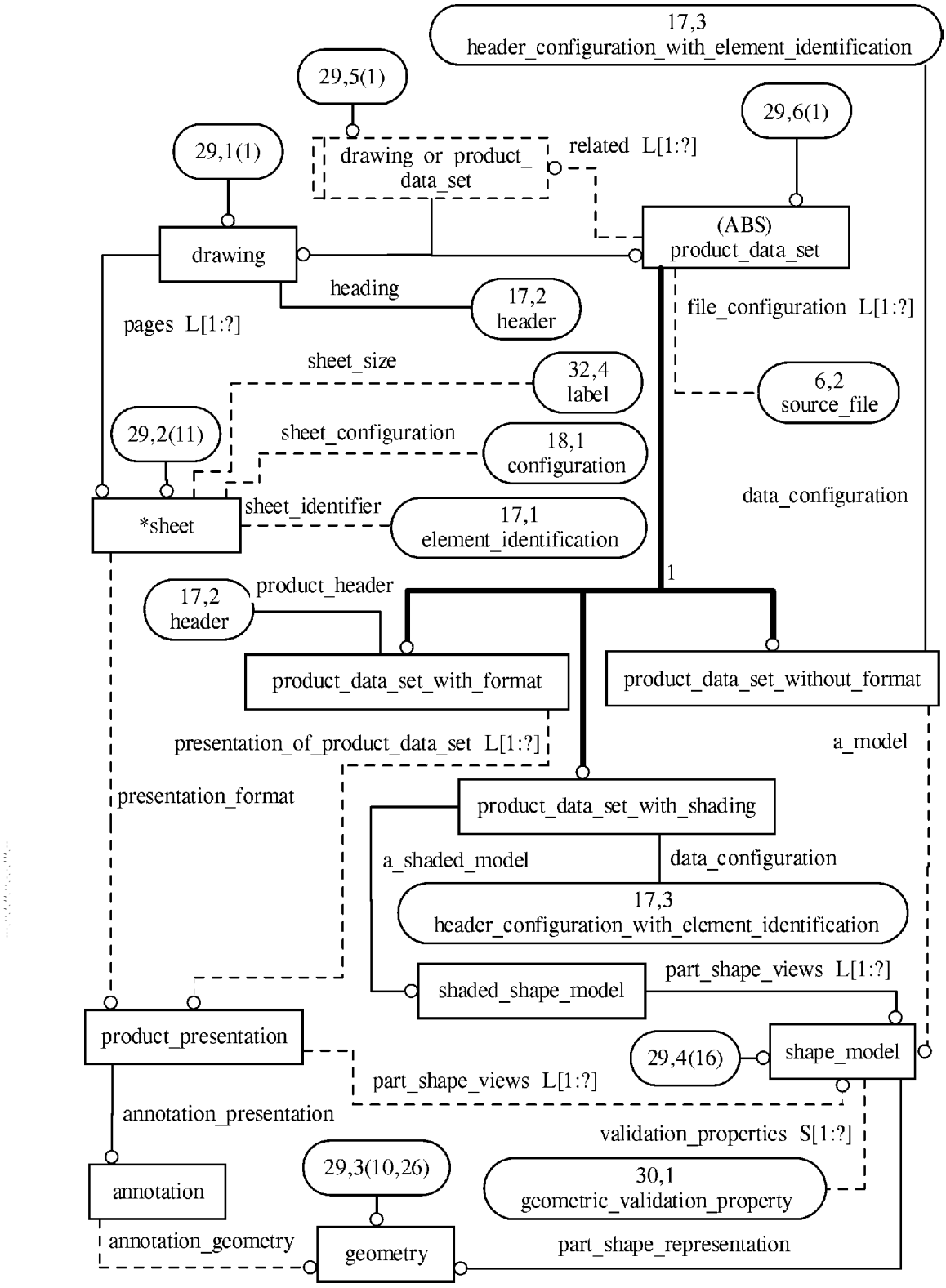

Figure G.29 — ARM diagram (29 of 33)

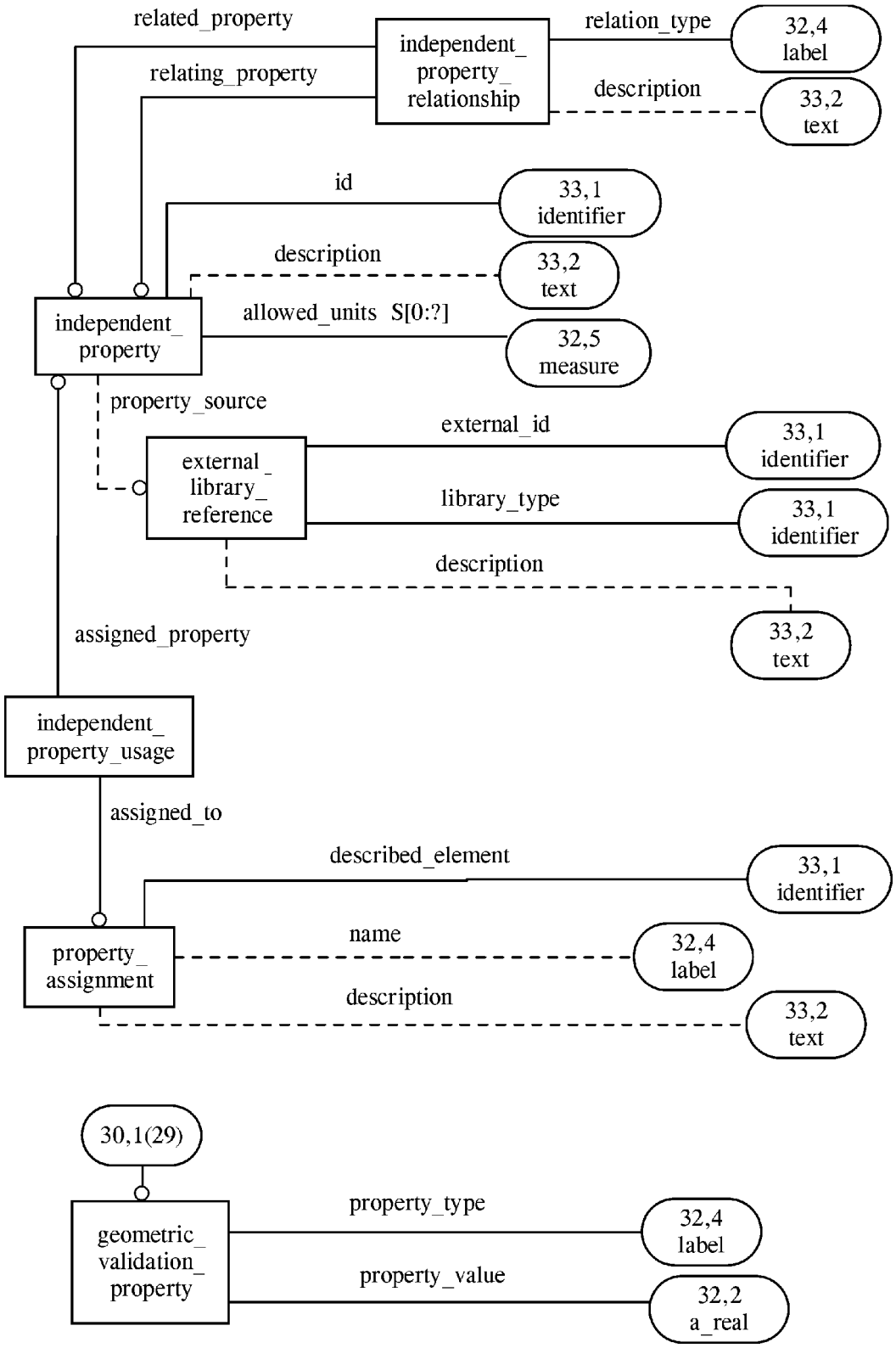

Figure G.30 — ARM diagram (30 of 33)

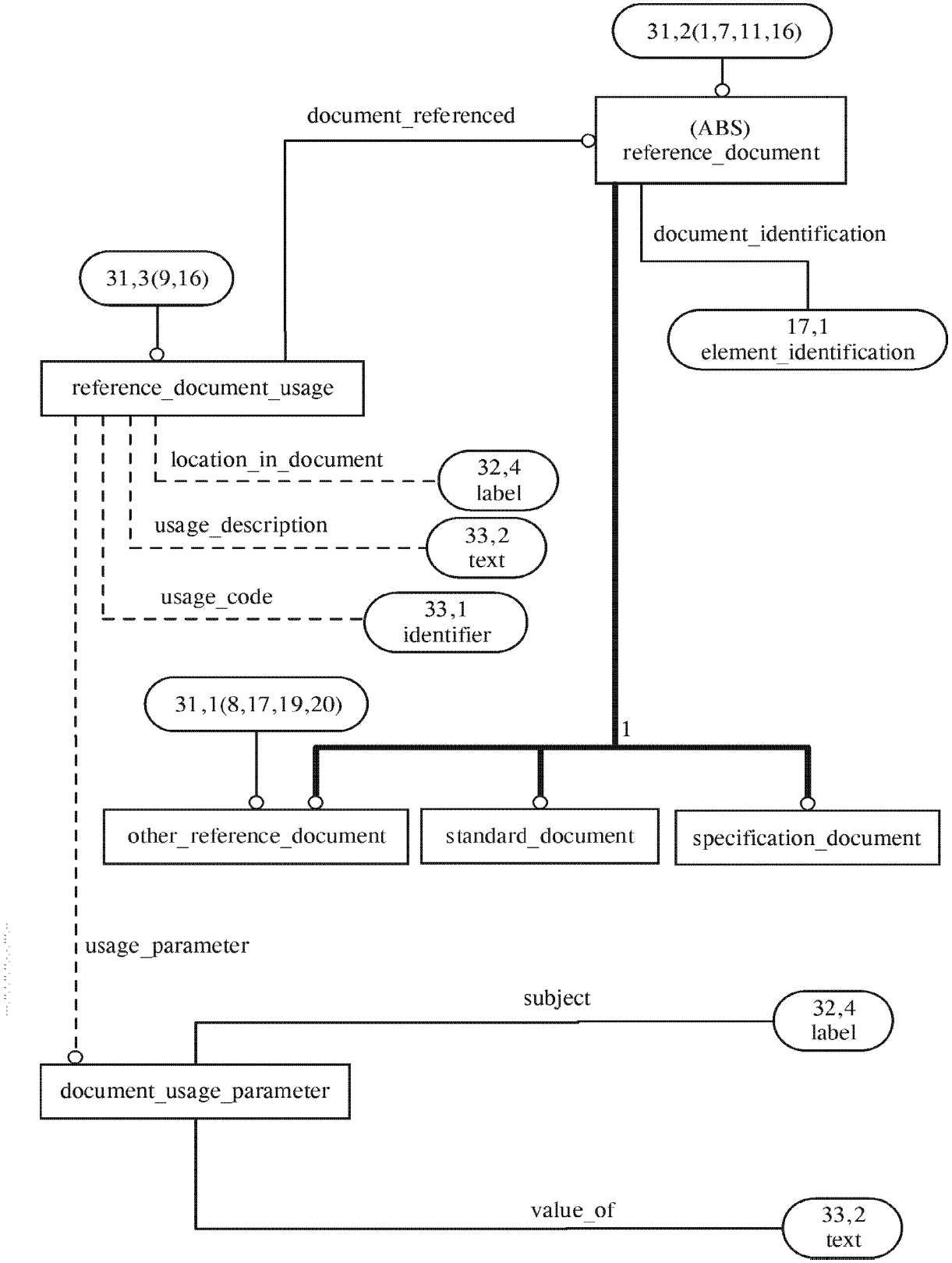

Figure G.31 — ARM diagram (31 of 33)

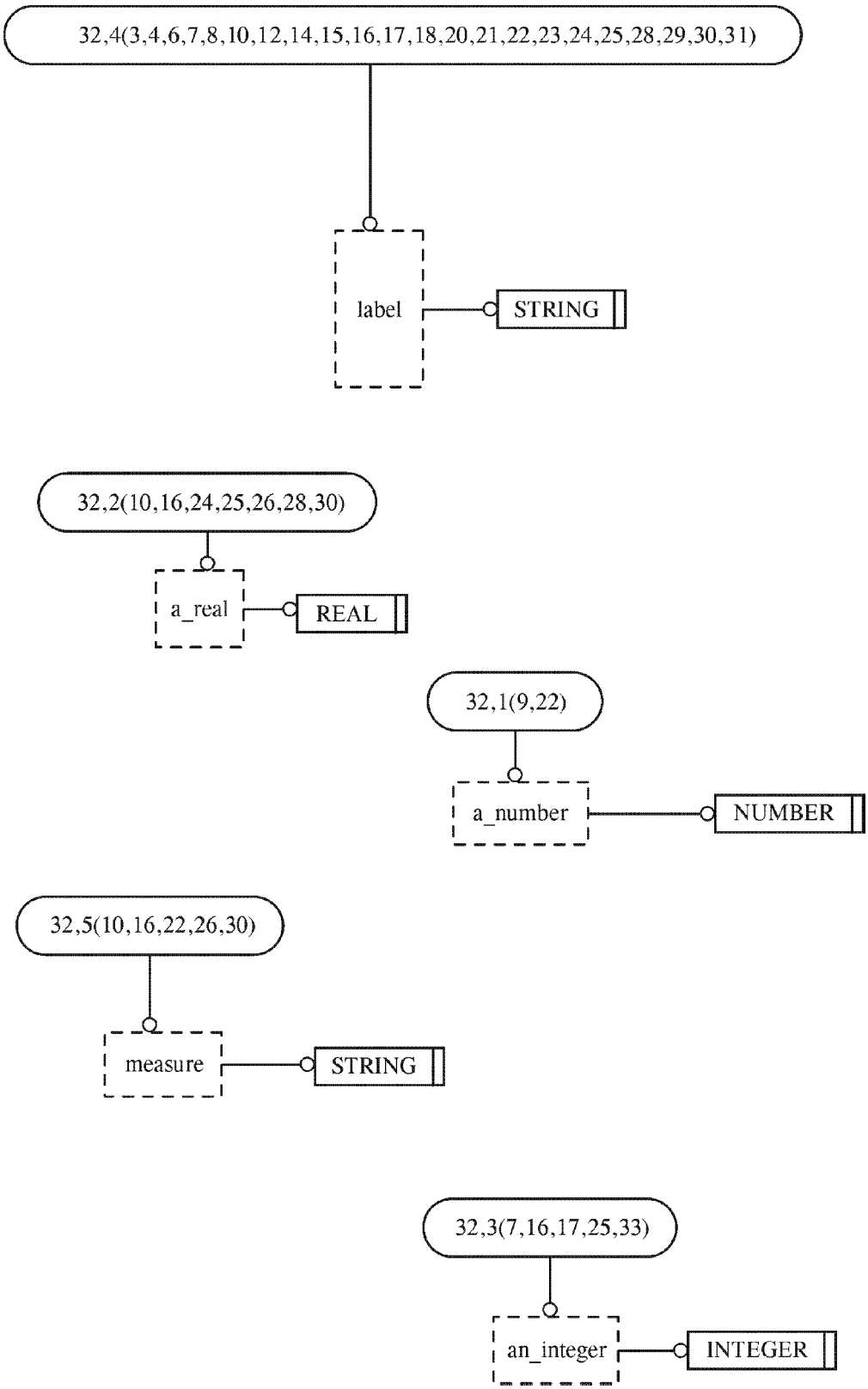

Figure G.32 — ARM diagram (32 of 33)

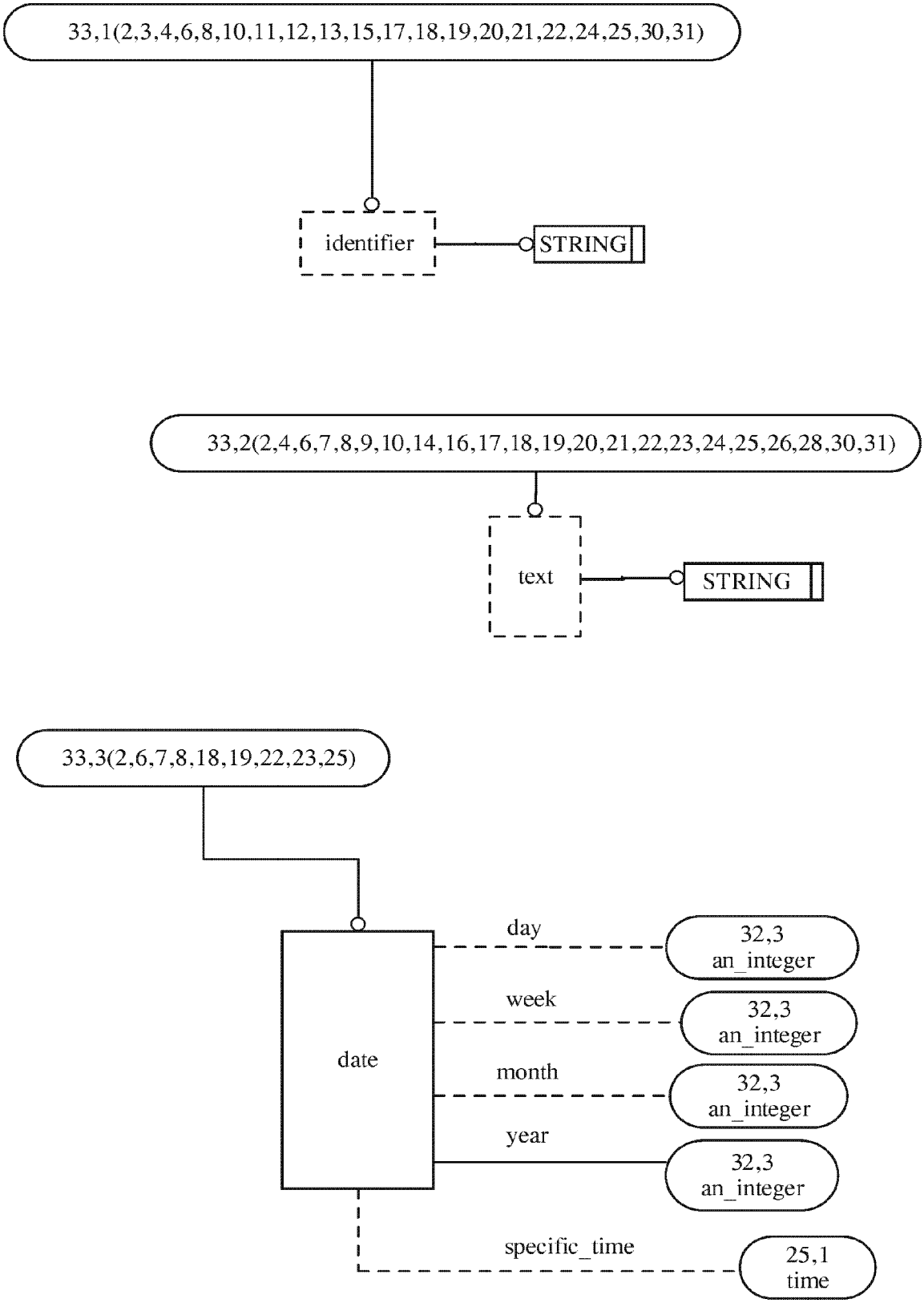

Figure G.33 — ARM diagram (33 of 33)

Annex H
(informative)

AIM EXPRESS-G

Figures H.1 through H.56 correspond to the AIM EXPRESS expanded listing given in annex A. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

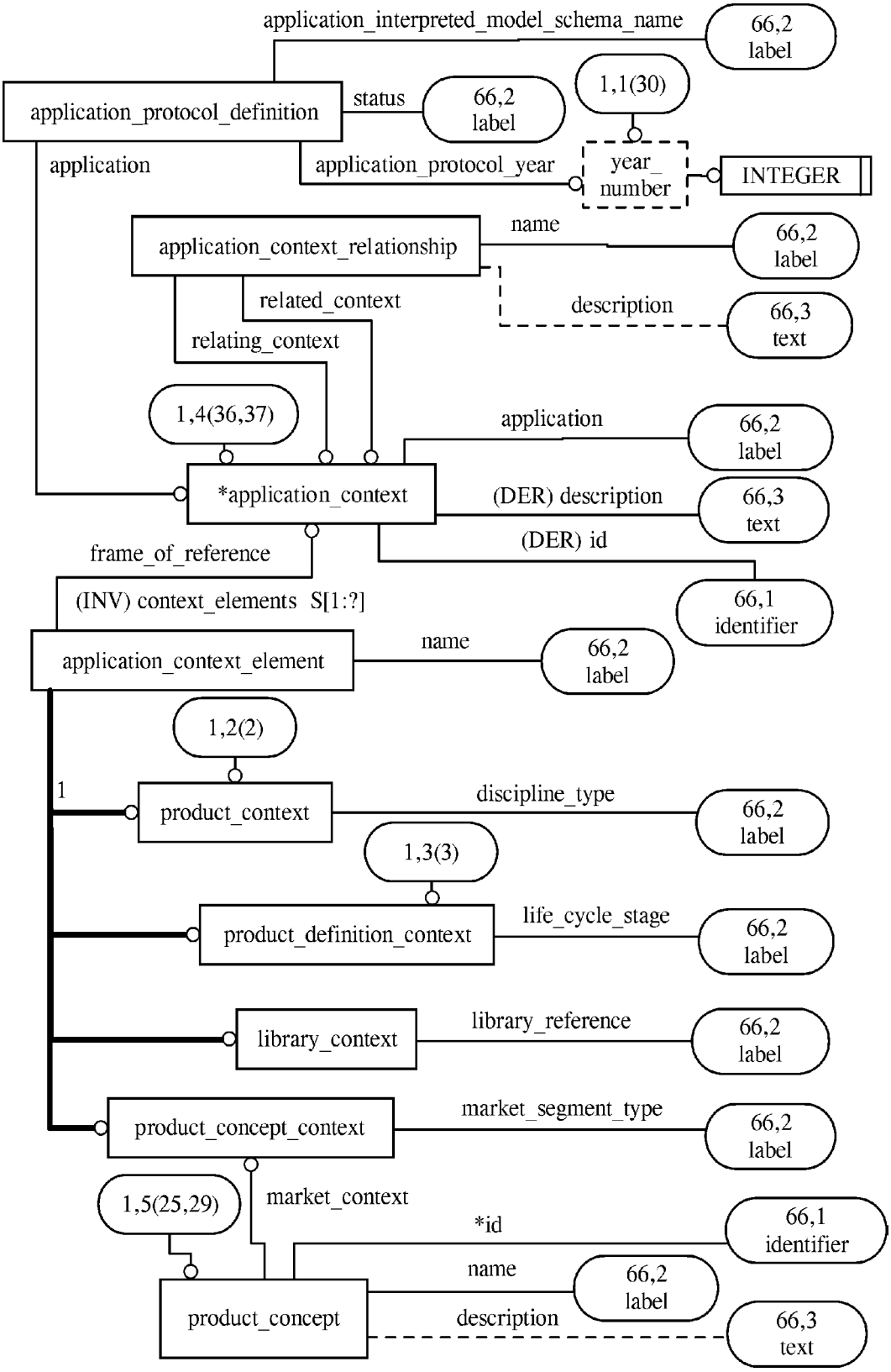

Figure H.1 — AIM EXPRESS-G diagram (1 of 66)

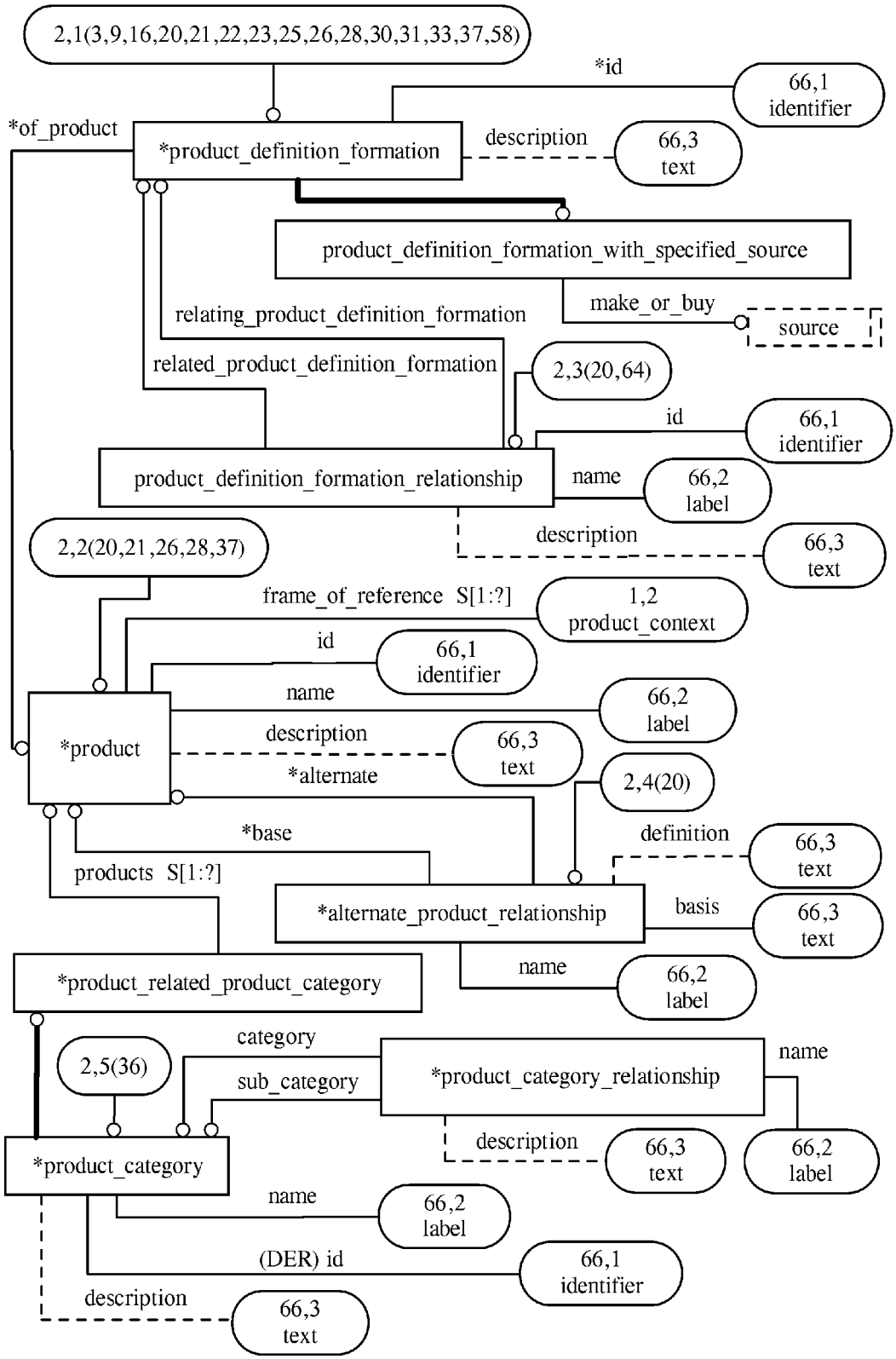

Figure H.2 — AIM EXPRESS-G diagram (2 of 66)

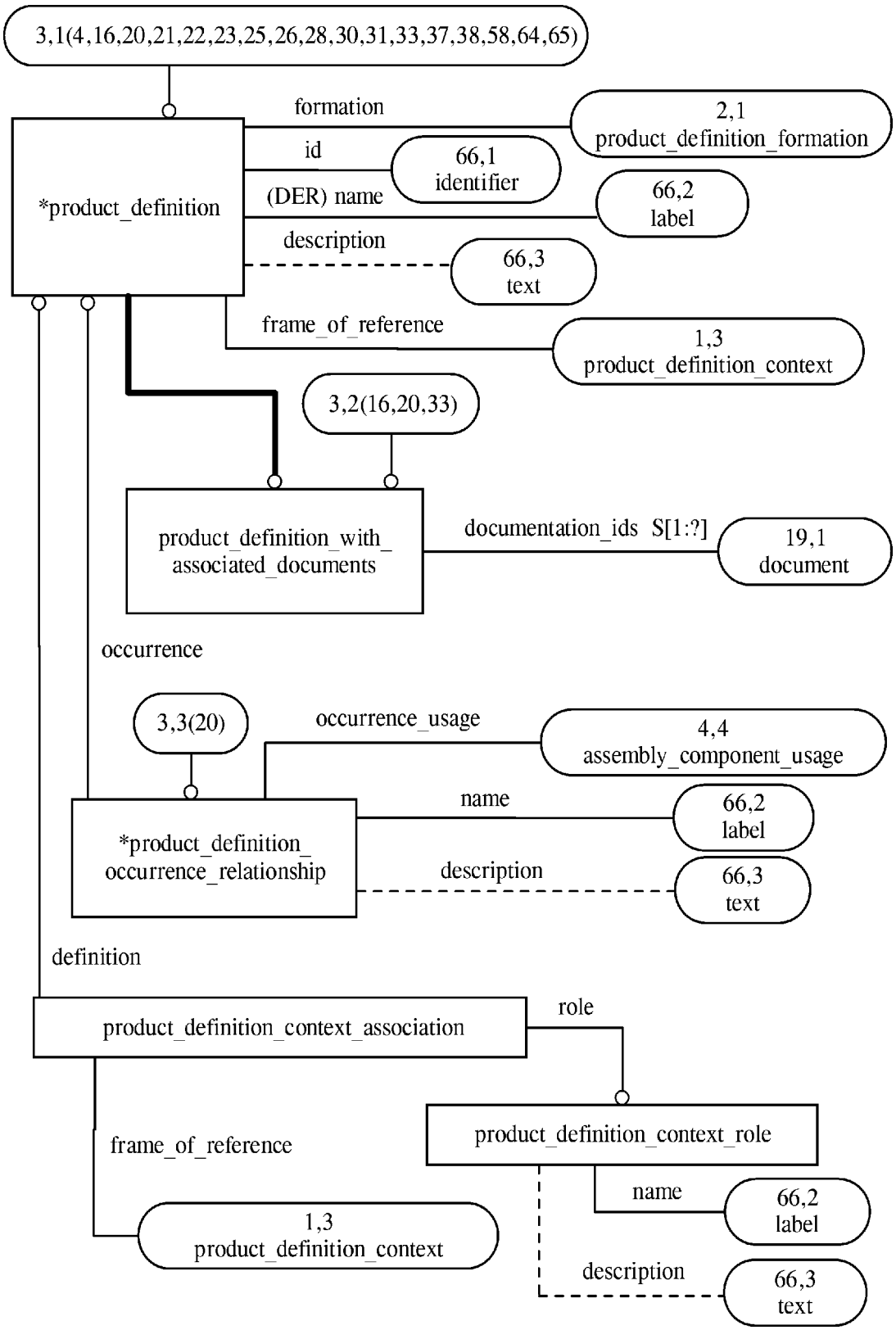

Figure H.3 — AIM EXPRESS-G diagram (3 of 66)

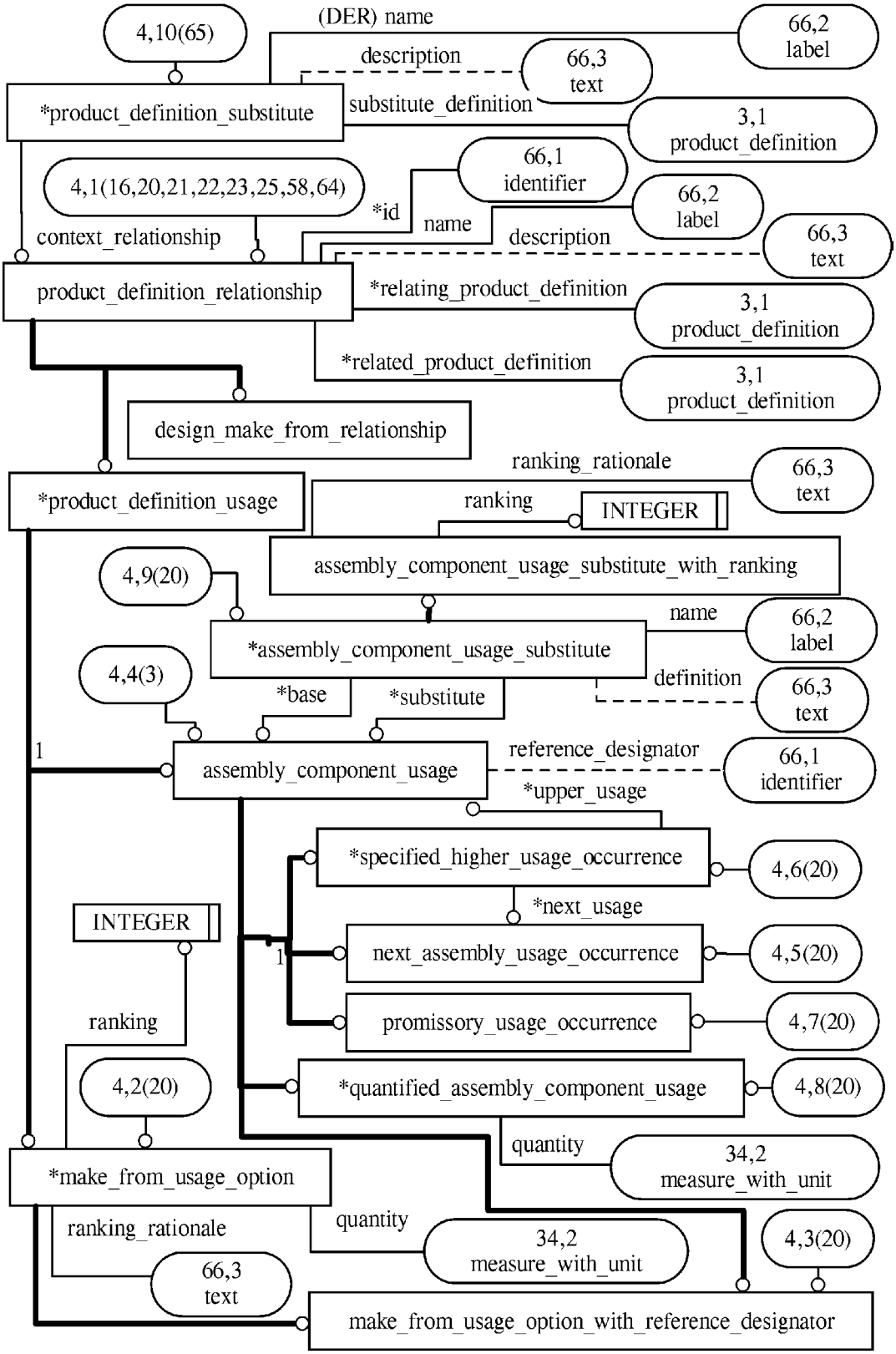

Figure H.4 — AIM EXPRESS-G diagram (4 of 66)

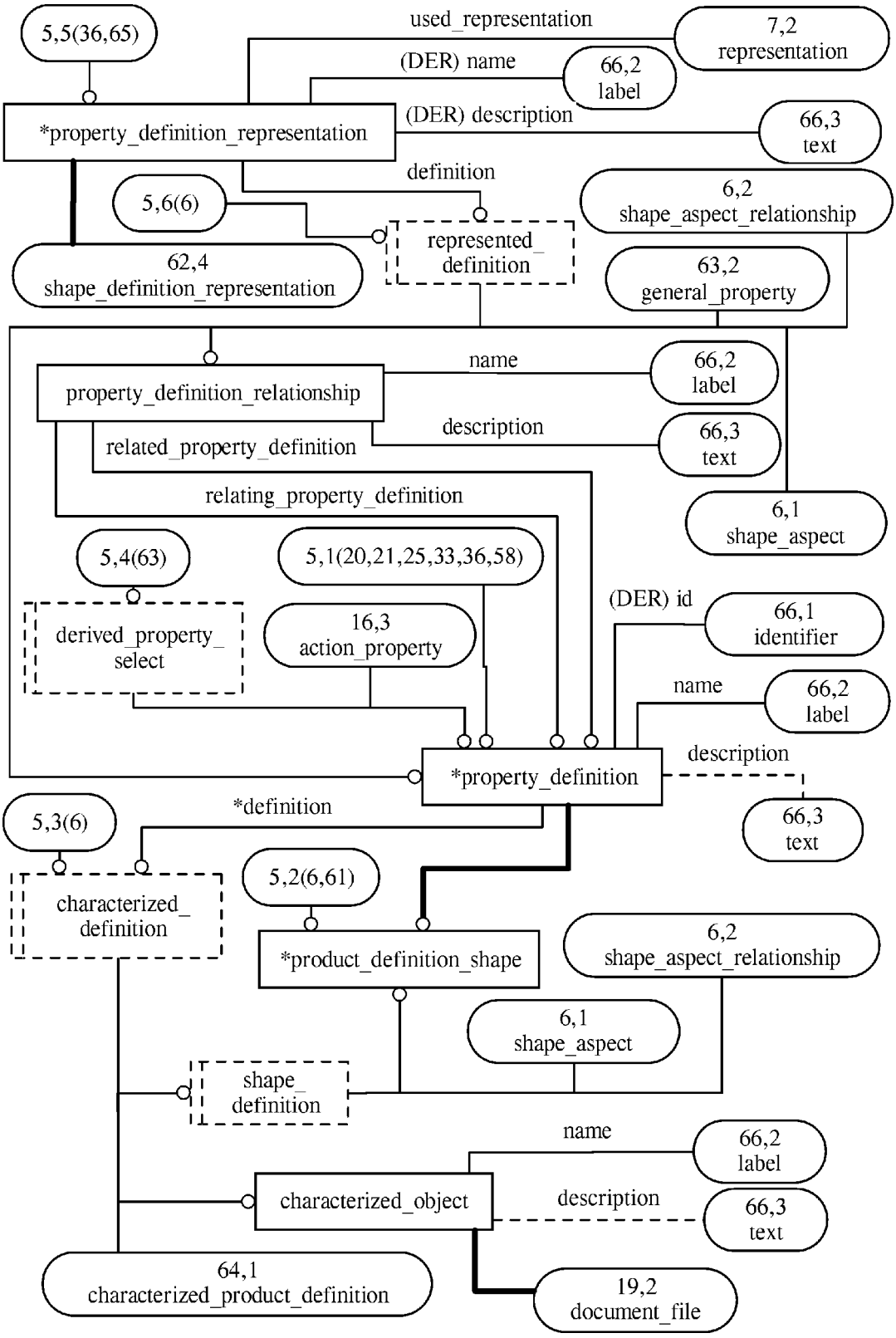

Figure H.5 — AIM EXPRESS-G diagram (5 of 66)

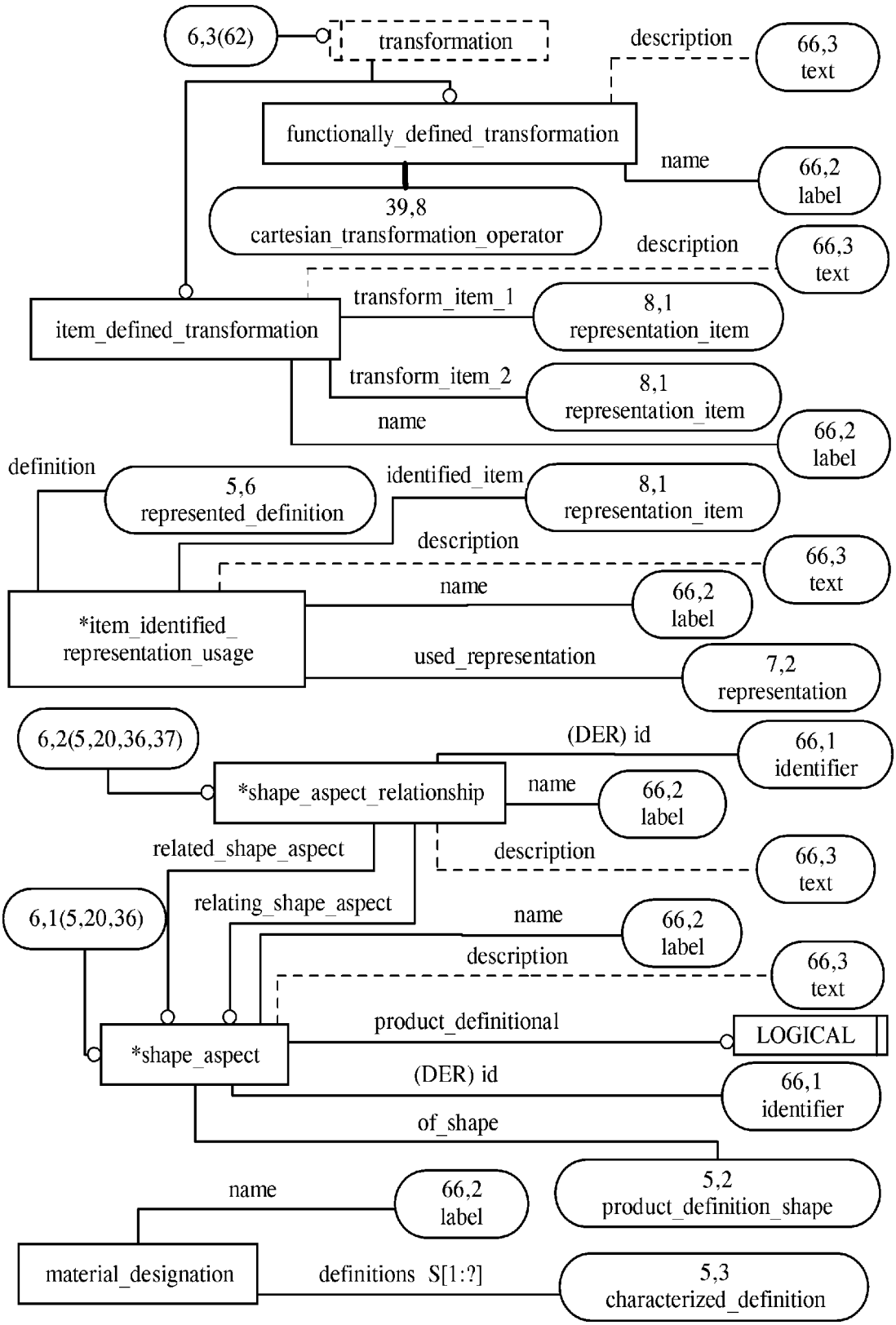

Figure H.6 — AIM EXPRESS-G diagram (6 of 66)

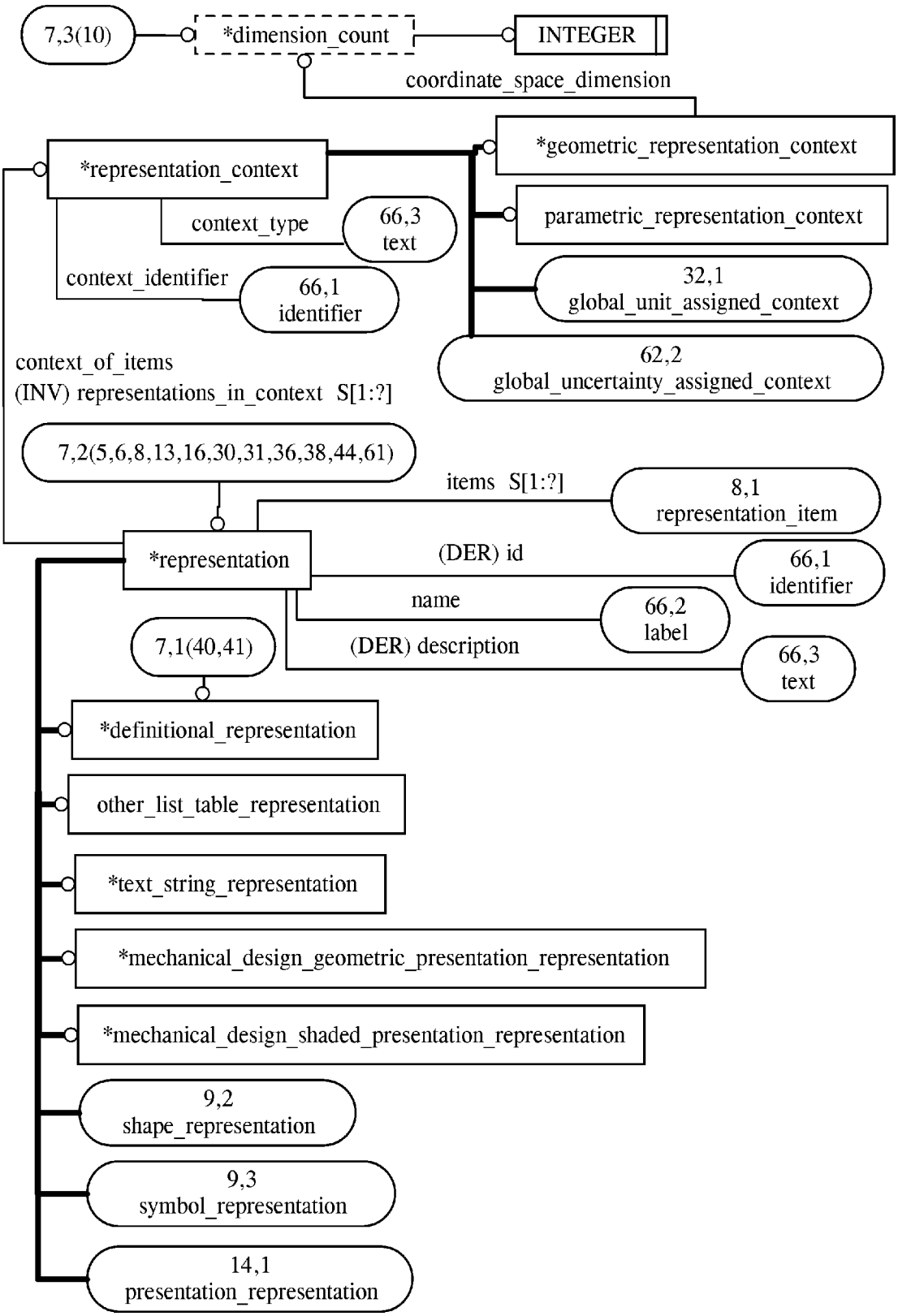

Figure H.7 — AIM EXPRESS-G diagram (7 of 66)

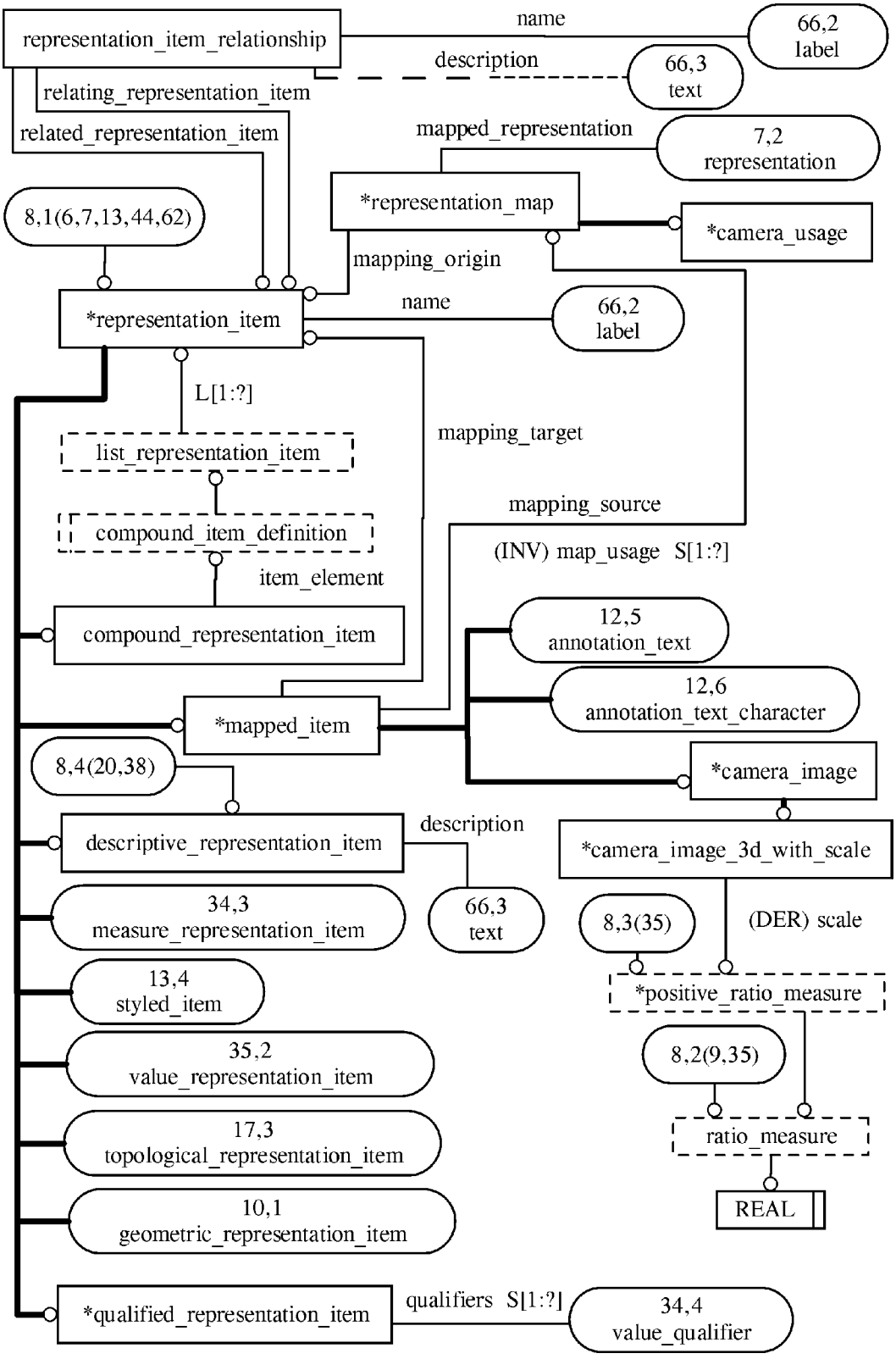

Figure H.8 — AIM EXPRESS-G diagram (8 of 66)

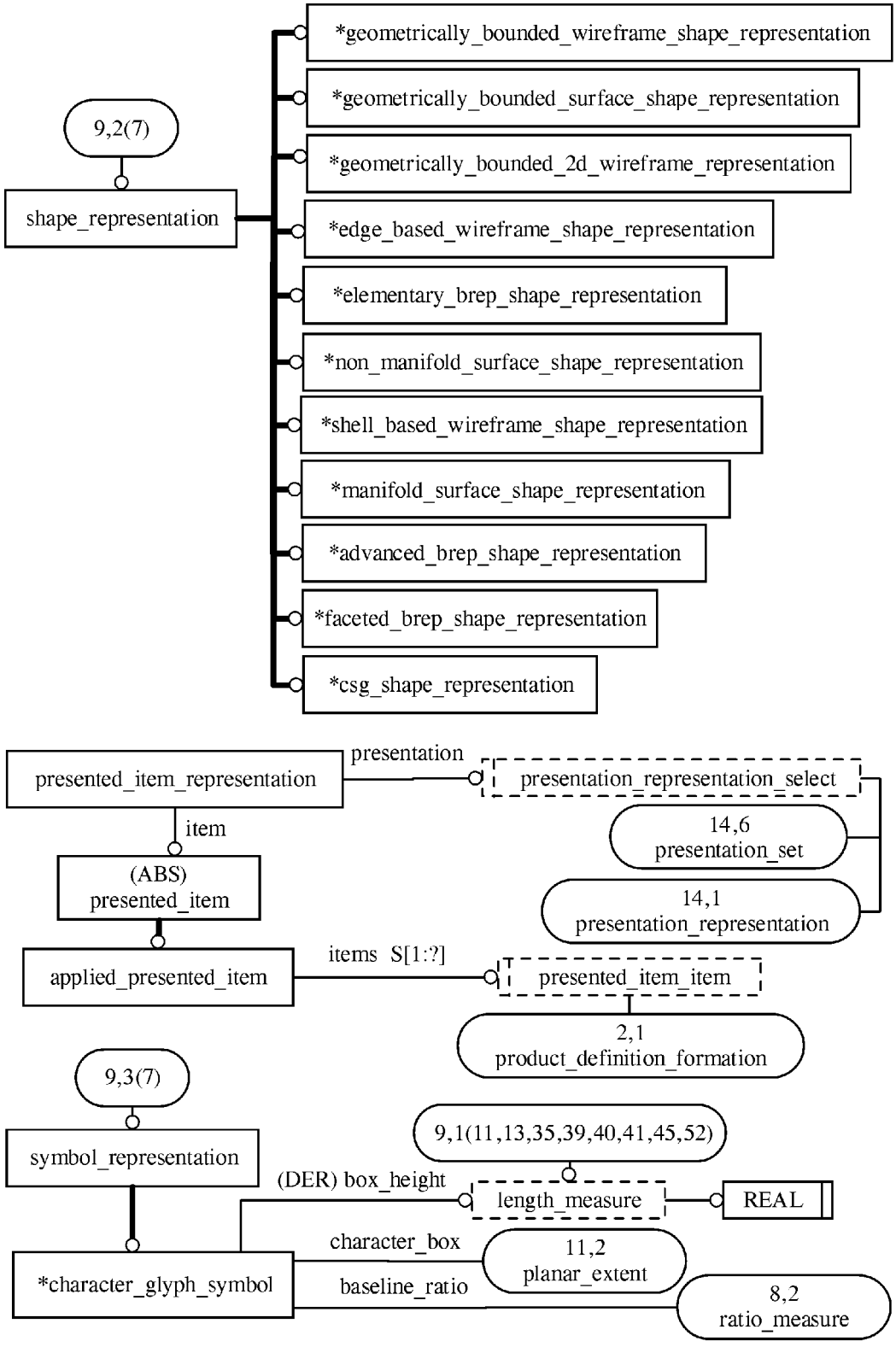

Figure H.9 — AIM EXPRESS-G diagram (9 of 66)

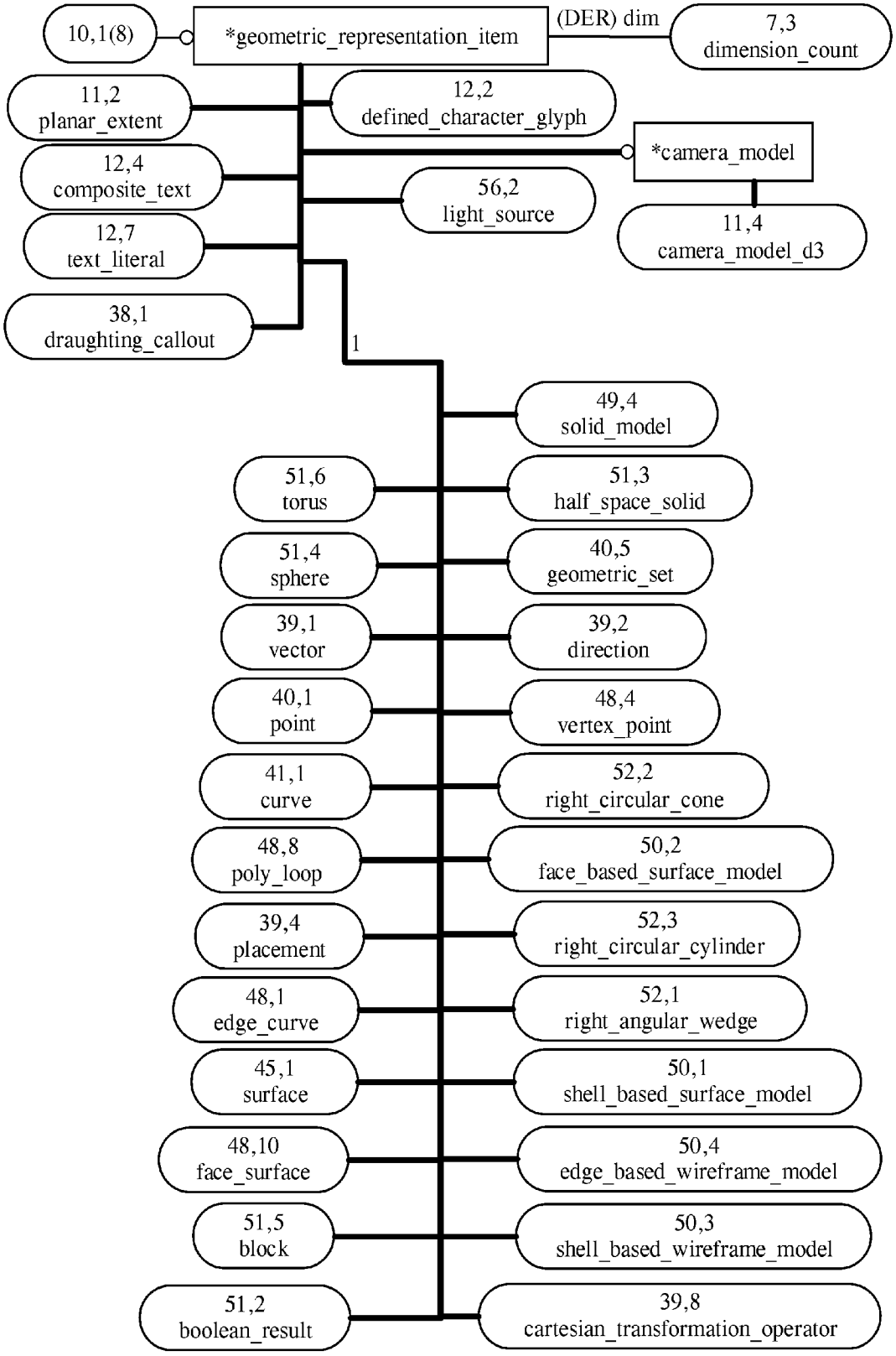

Figure H.10 — AIM EXPRESS-G diagram (10 of 66)

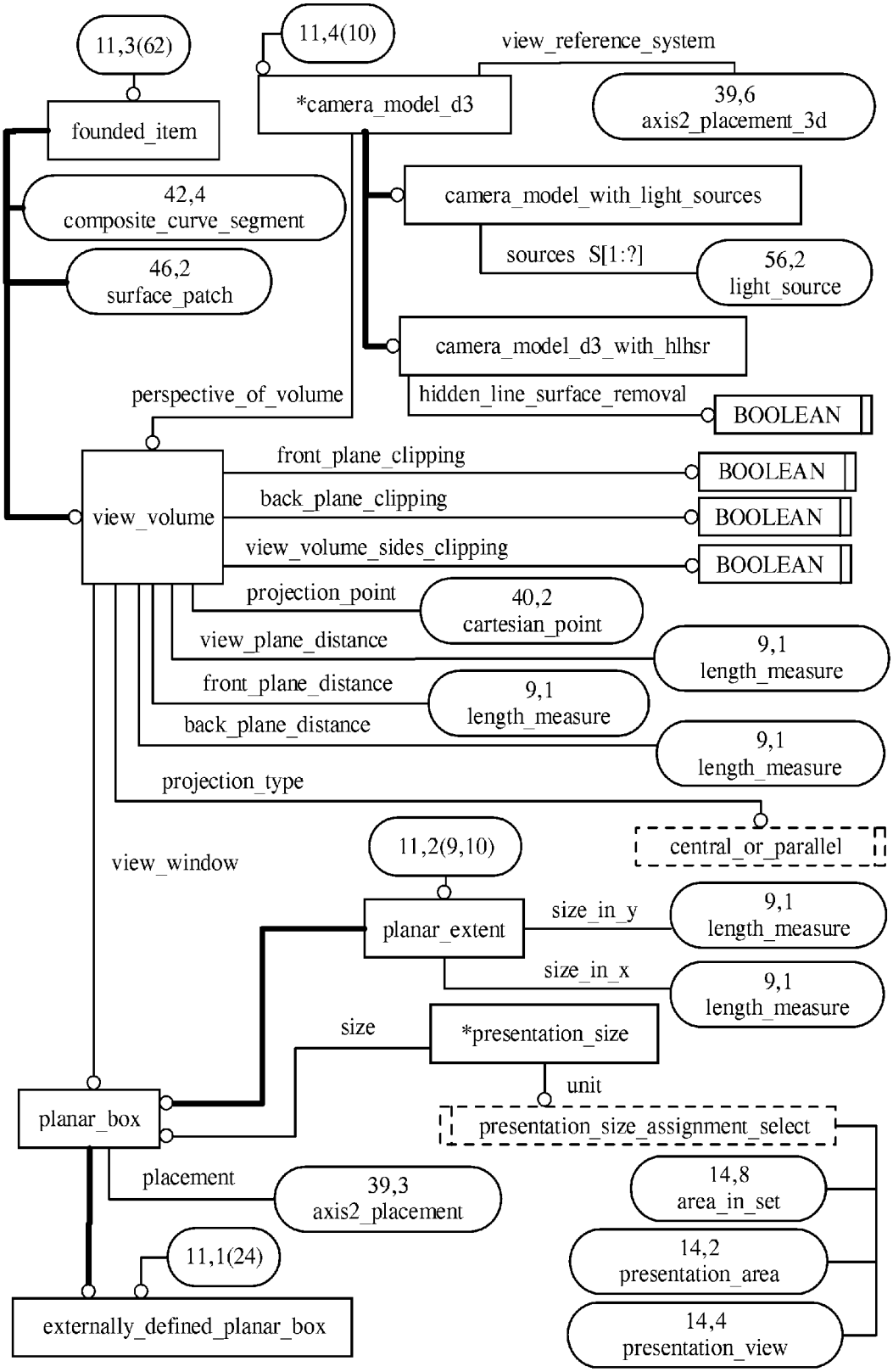

Figure H.11 — AIM EXPRESS-G diagram (11 of 66)

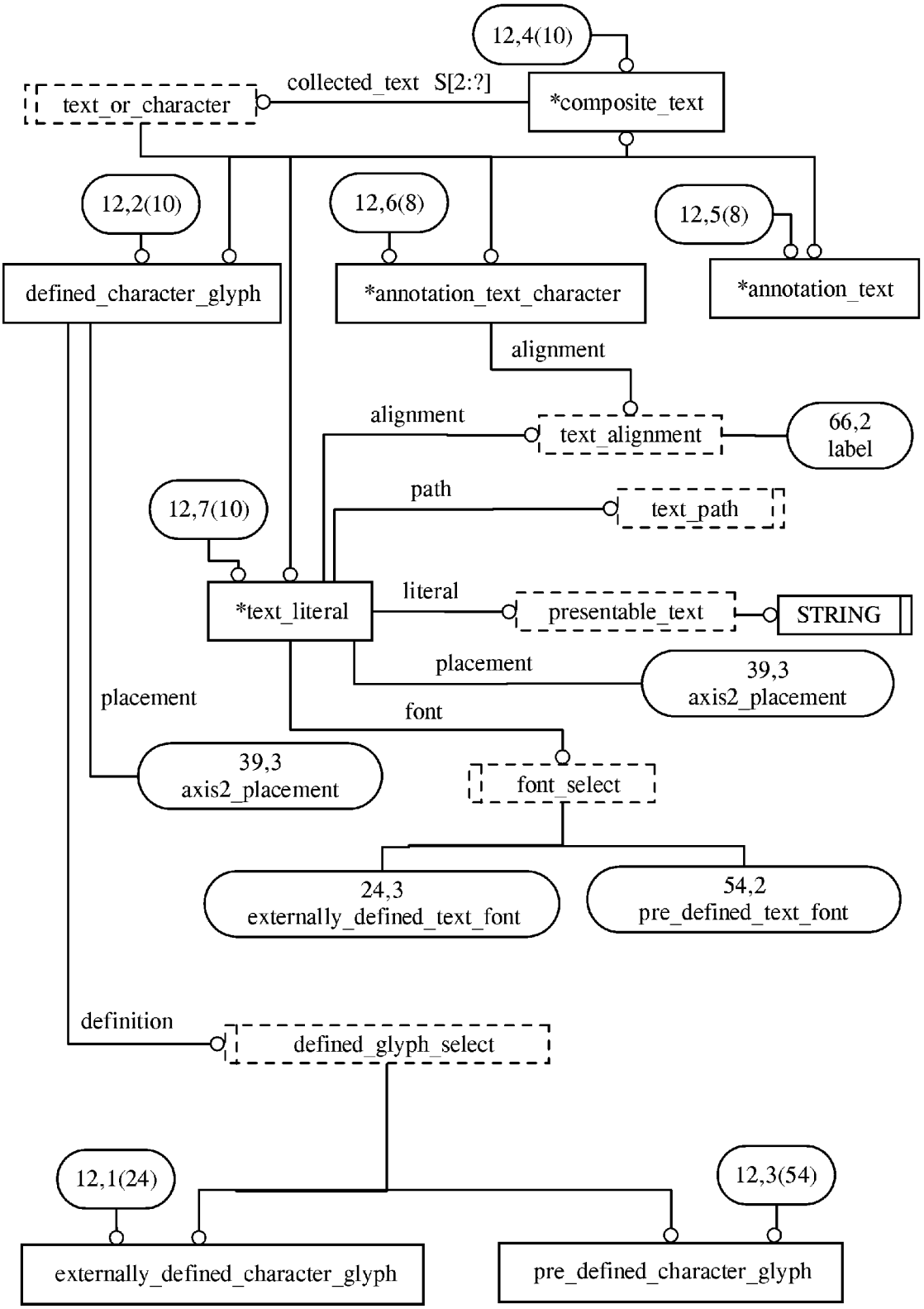

Figure H.12 — AIM EXPRESS-G diagram (12 of 66)

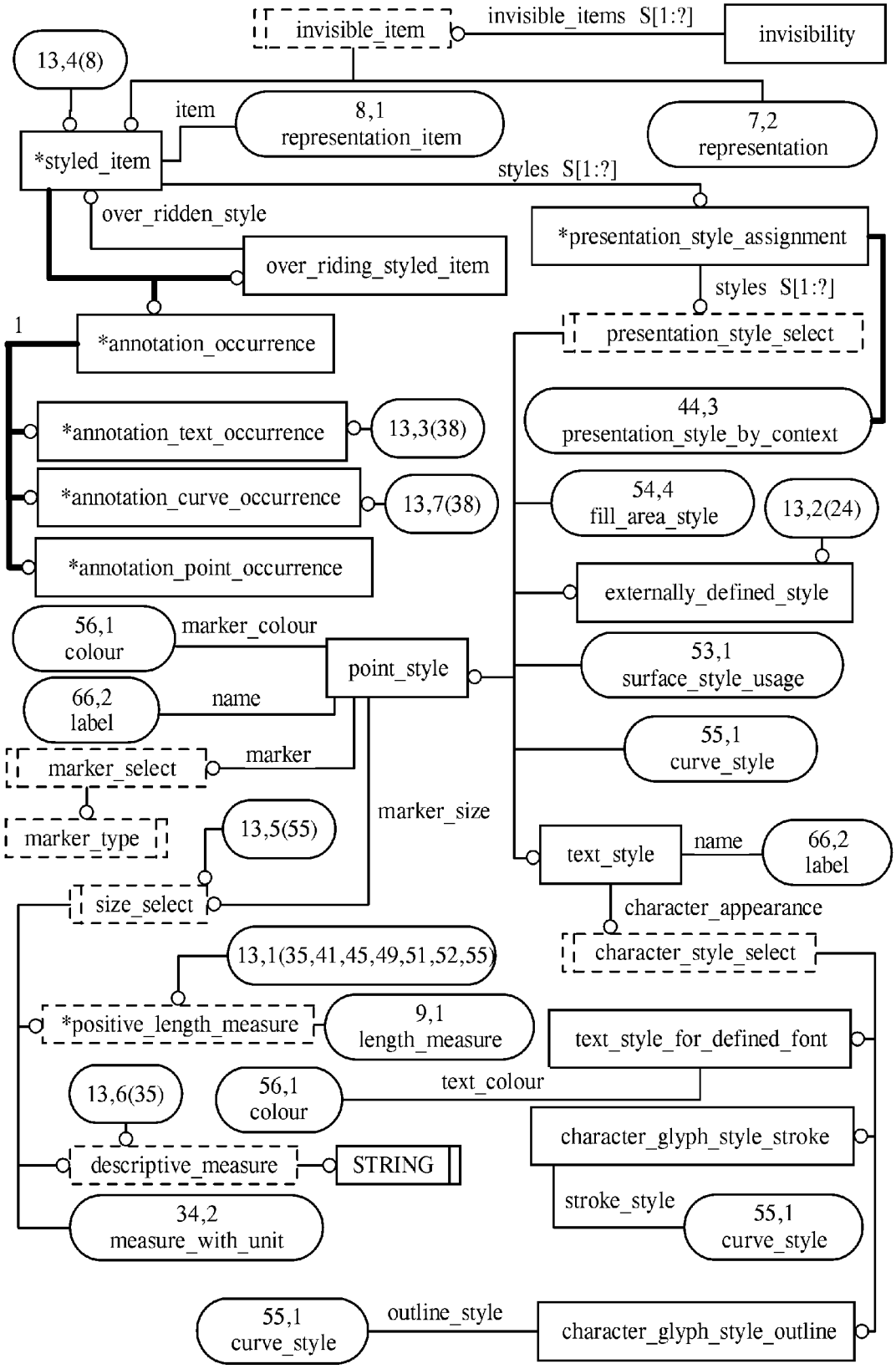

Figure H.13 — AIM EXPRESS-G diagram (13 of 66)

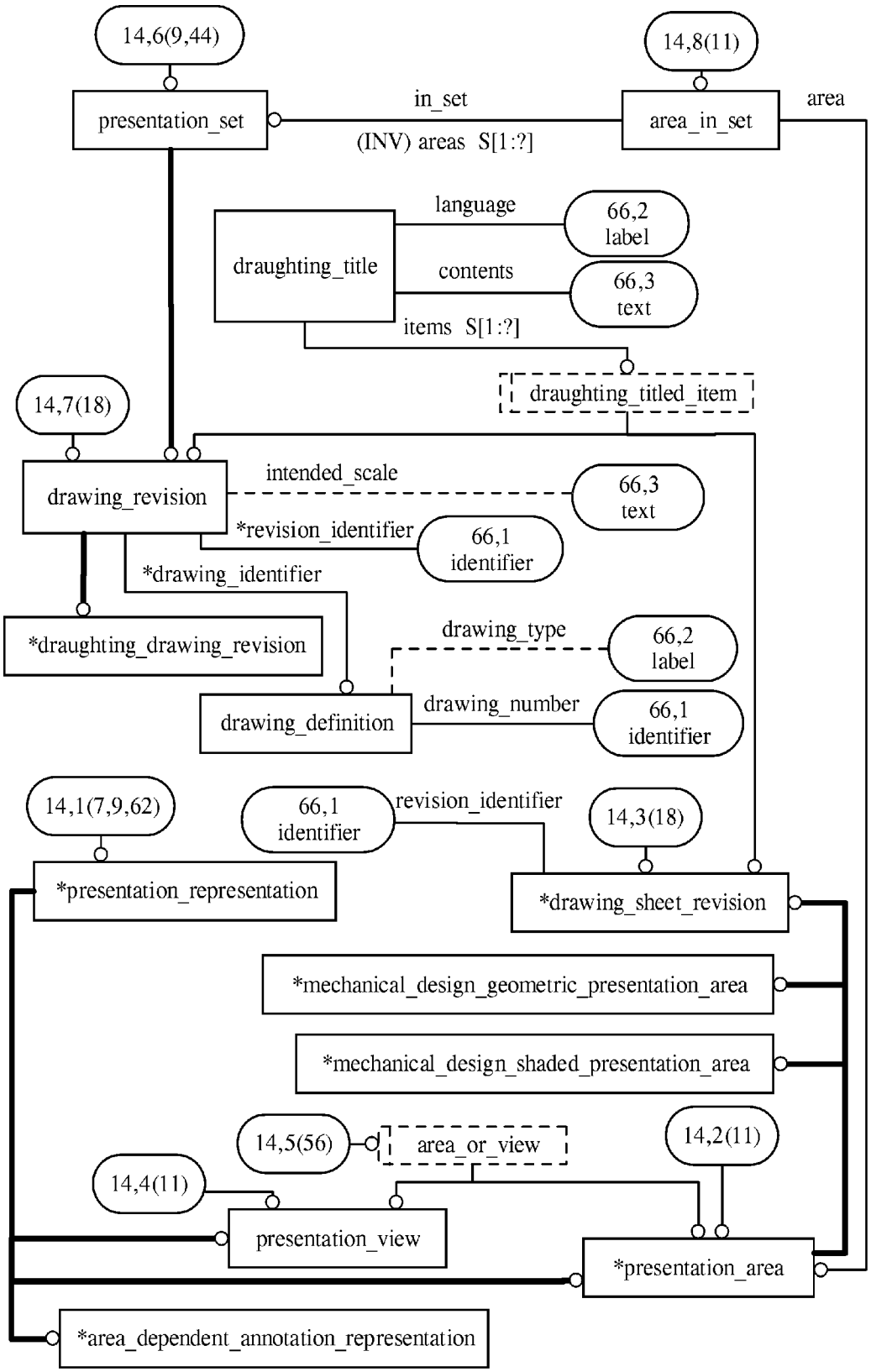

Figure H.14 — AIM EXPRESS-G diagram (14 of 66)

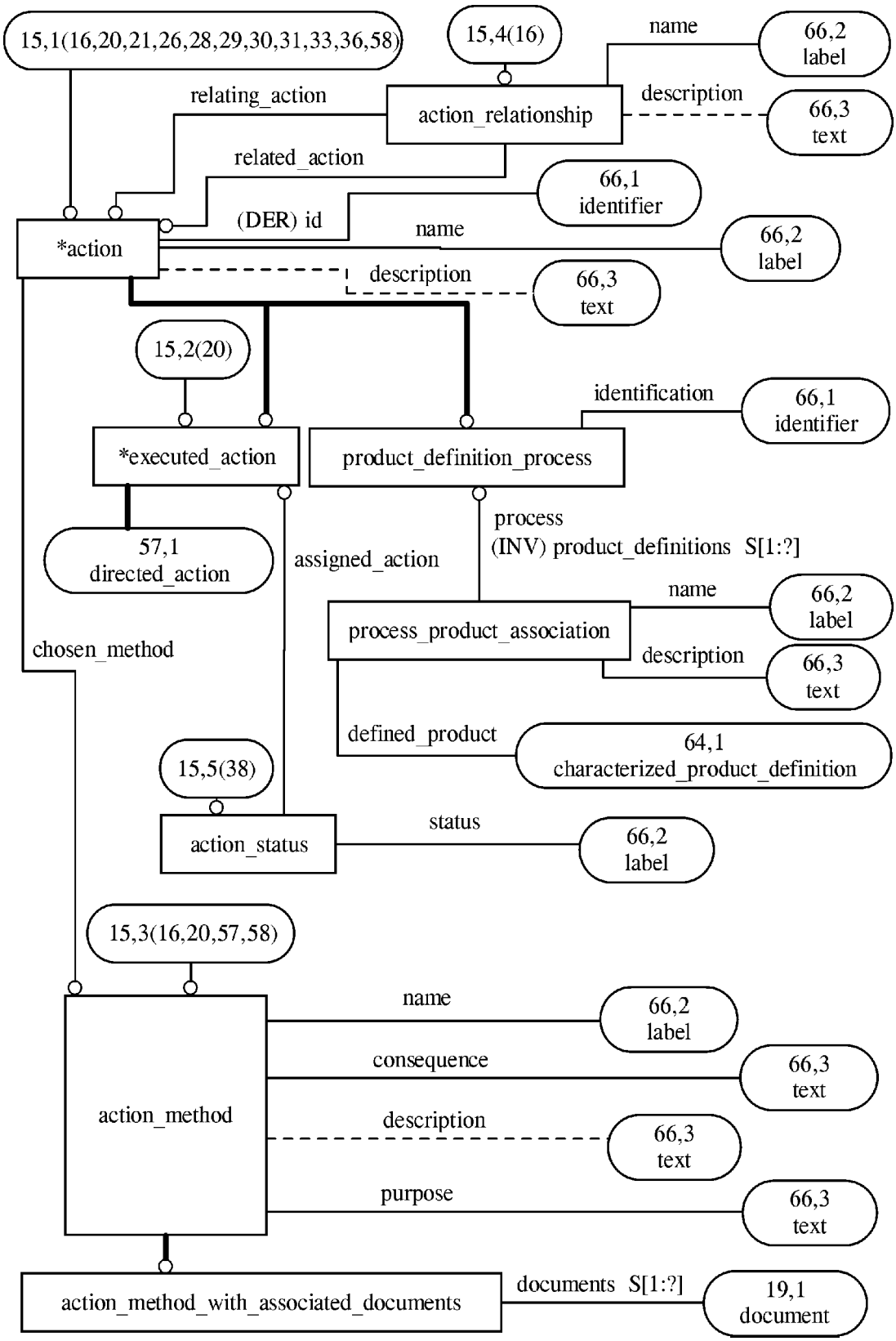

Figure H.15 — AIM EXPRESS-G diagram (15 of 66)

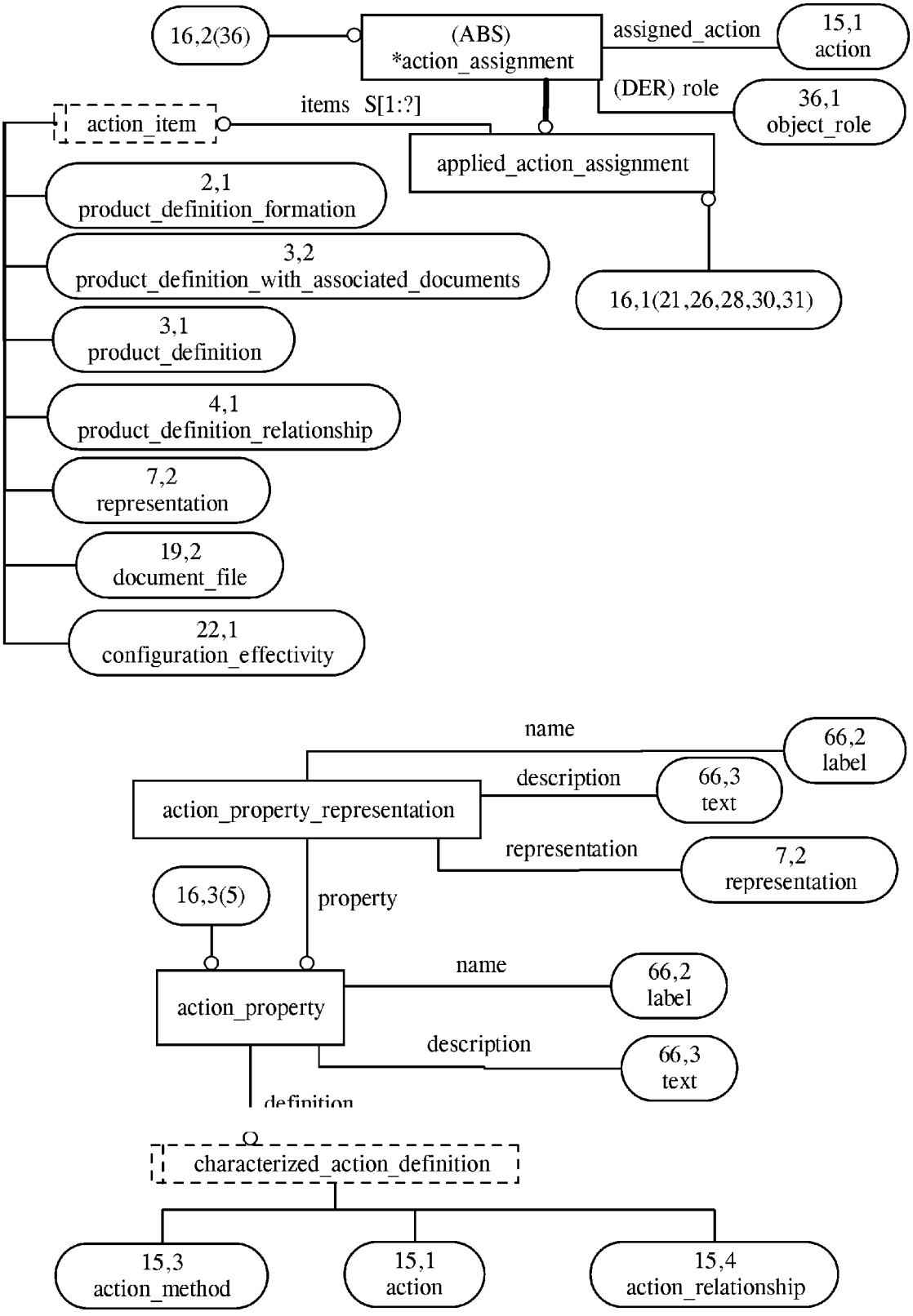

Figure H.16 — AIM EXPRESS-G diagram (16 of 66)

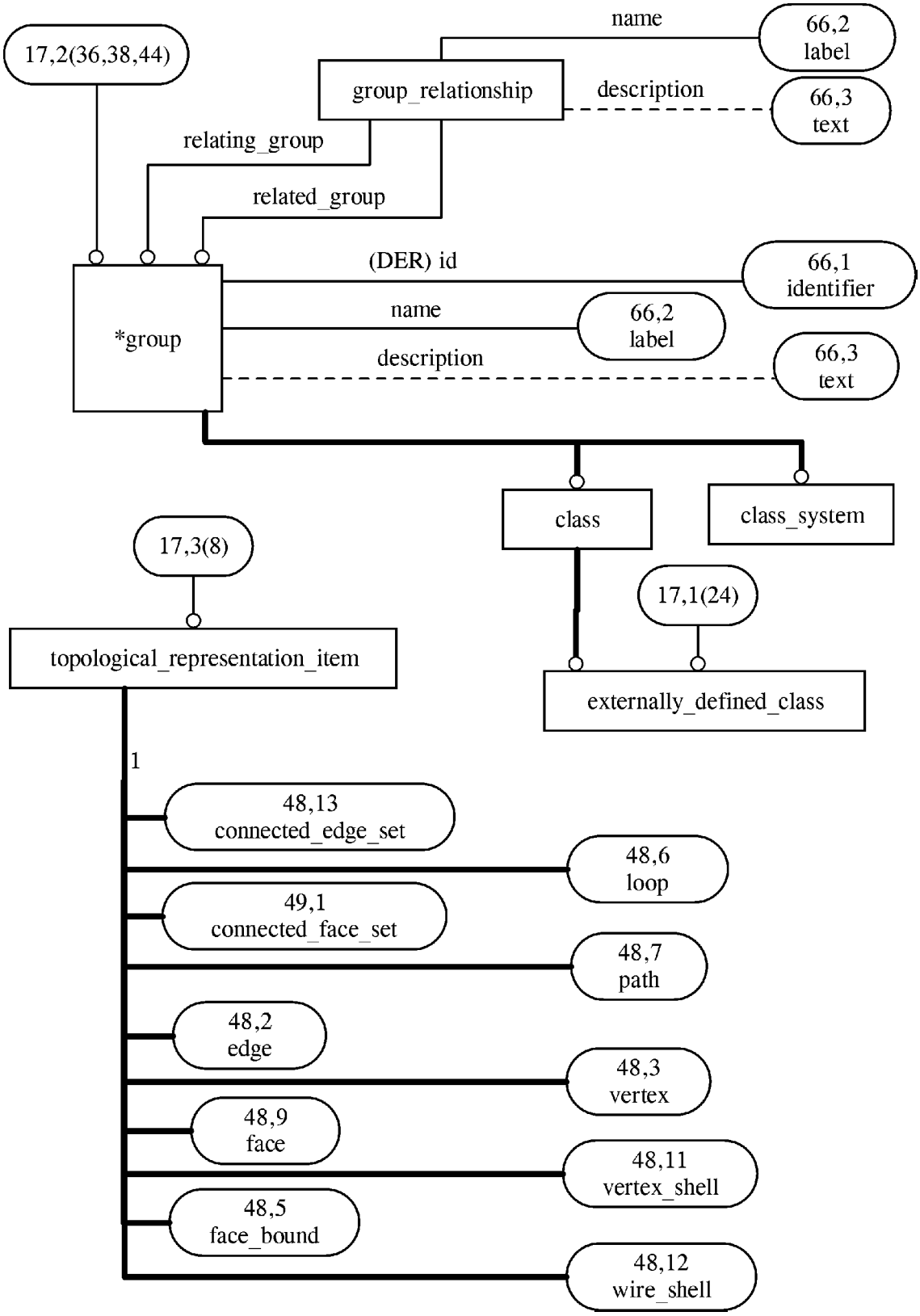

Figure H.17 — AIM EXPRESS-G diagram (17 of 66)

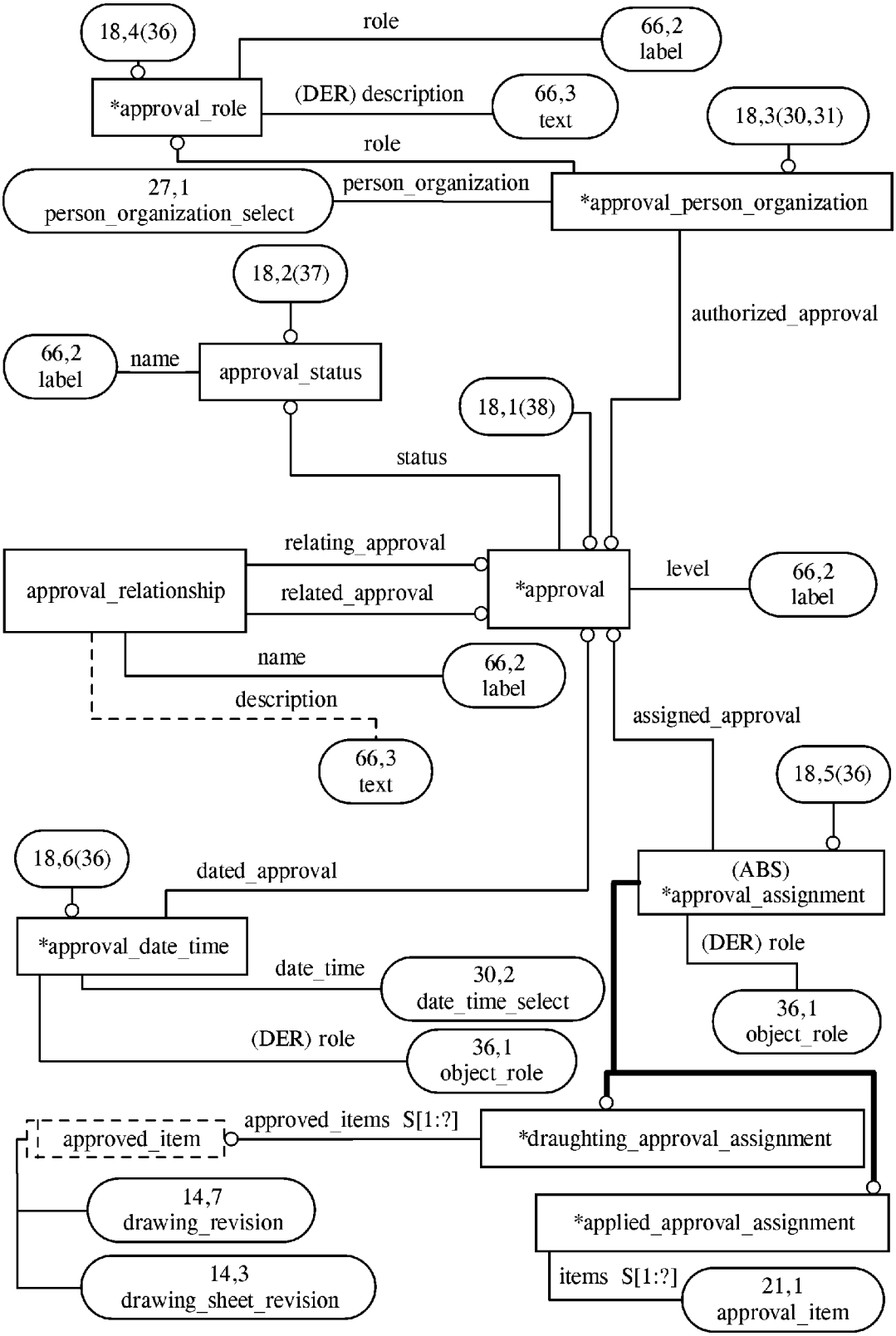

Figure H.18 — AIM EXPRESS-G diagram (18 of 66)

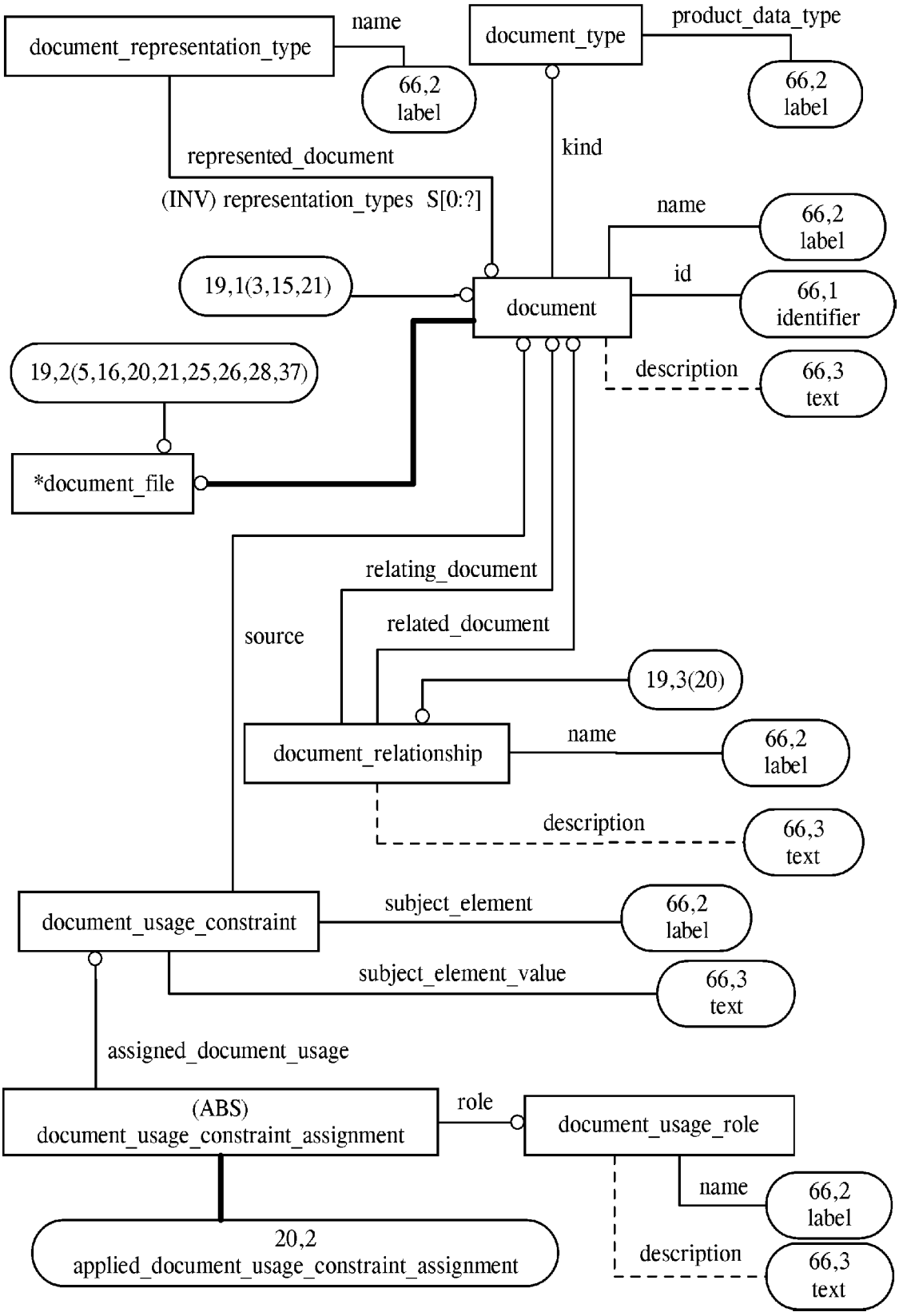

Figure H.19 — AIM EXPRESS-G diagram (19 of 66)

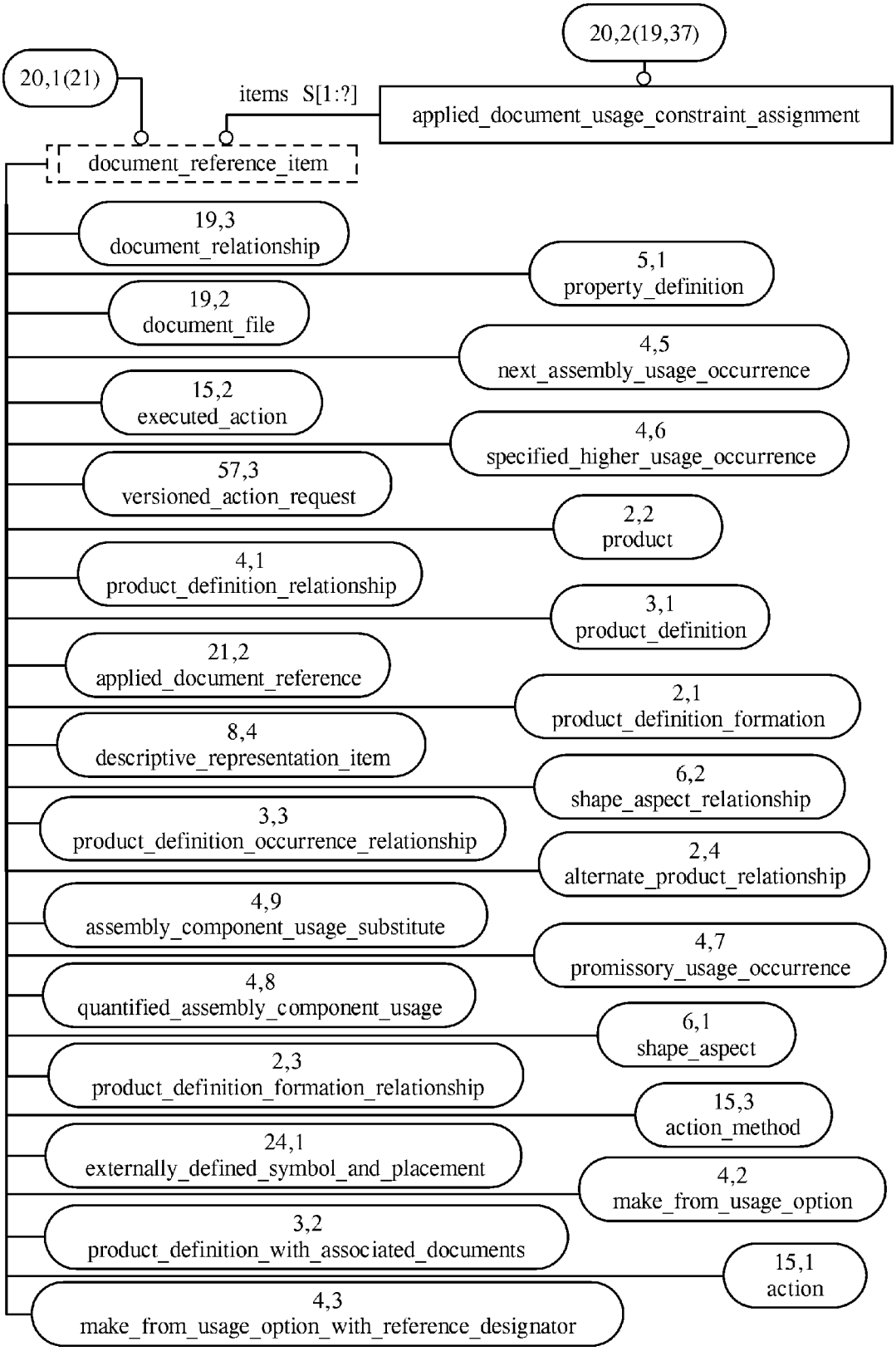

Figure H.20 — AIM EXPRESS-G diagram (20 of 66)

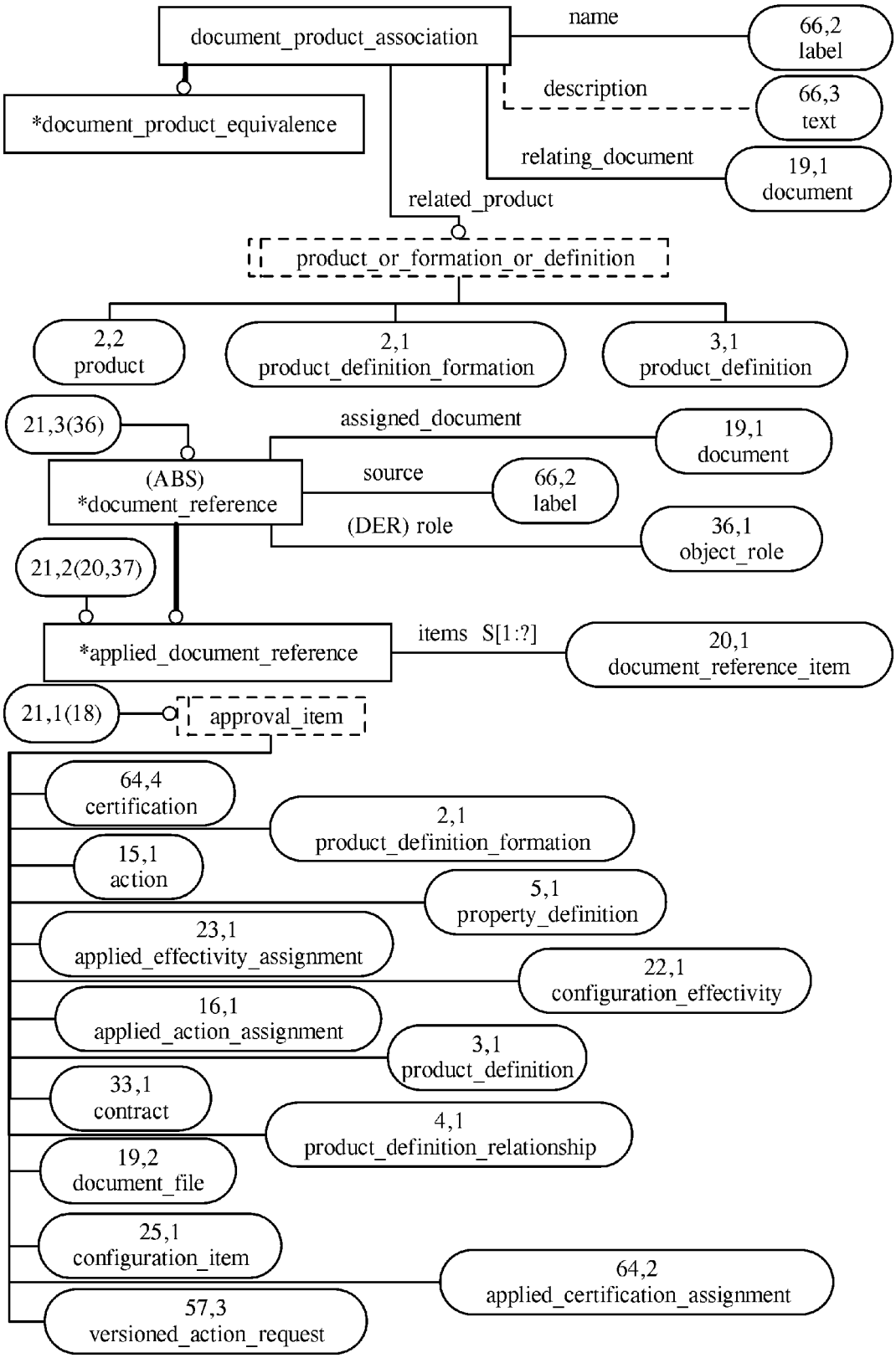

Figure H.21 — AIM EXPRESS-G diagram (21 of 66)

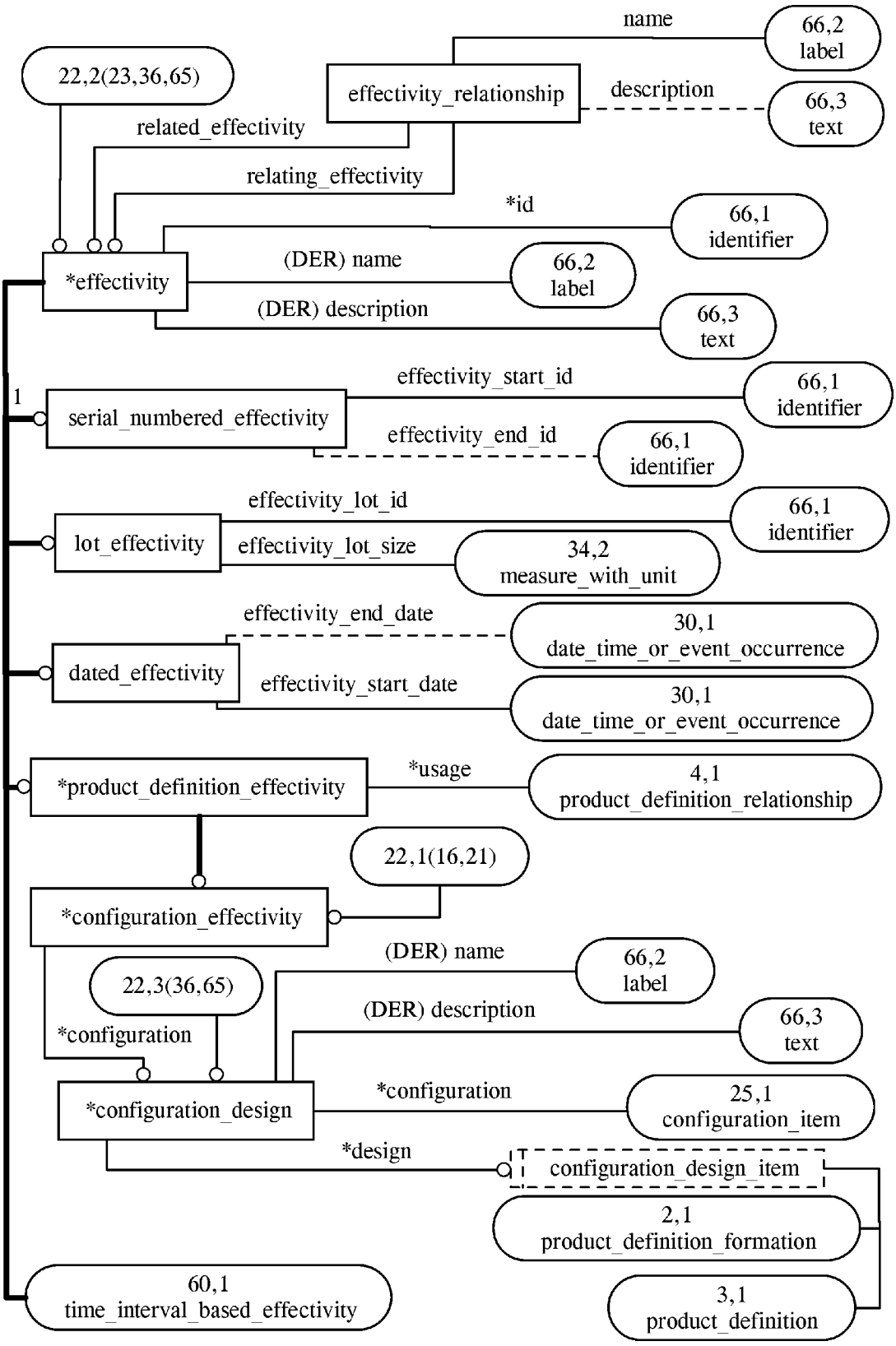

Figure H.22 — AIM EXPRESS-G diagram (22 of 66)

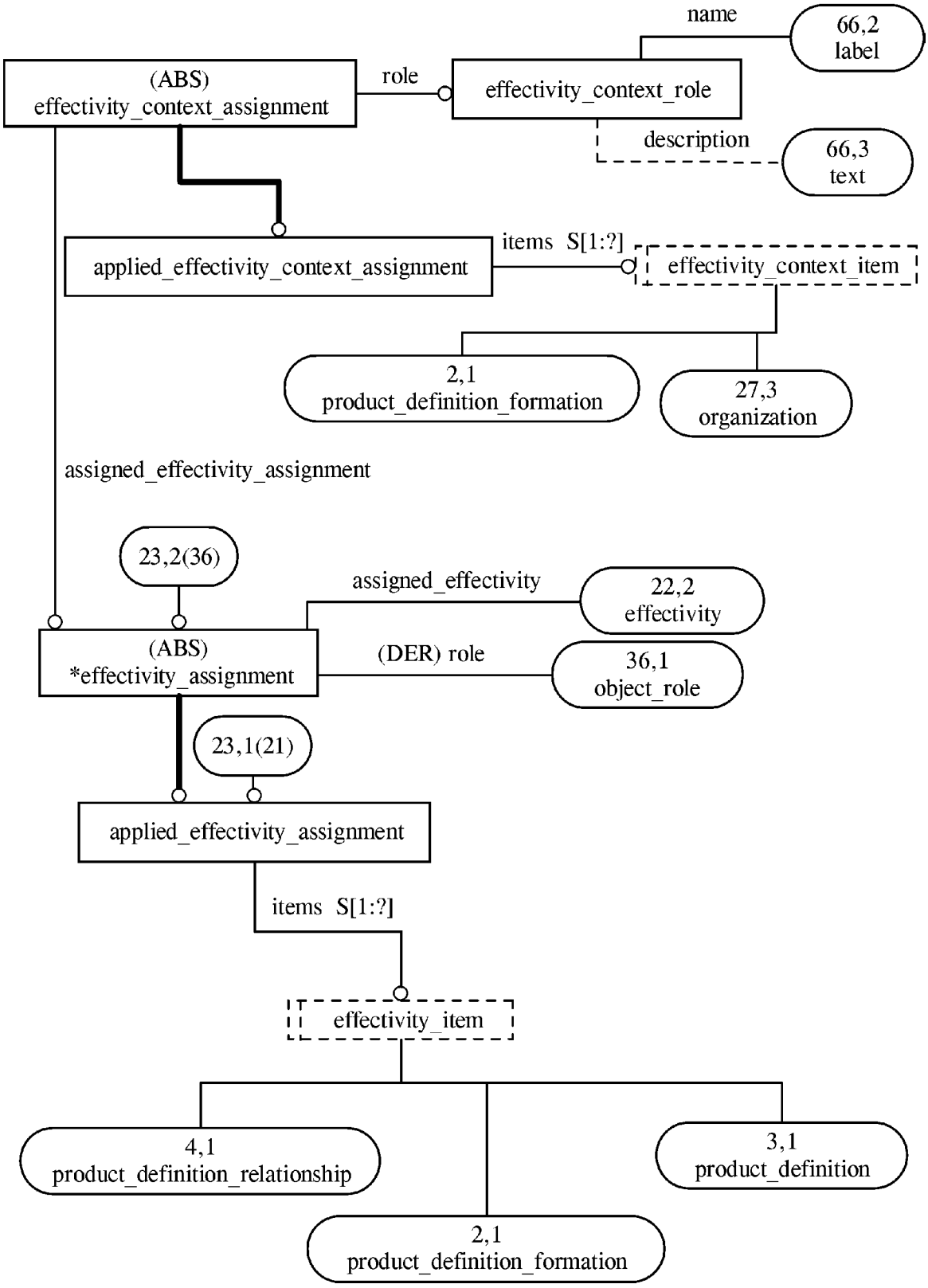

Figure H.23 — AIM EXPRESS-G diagram (23 of 66)

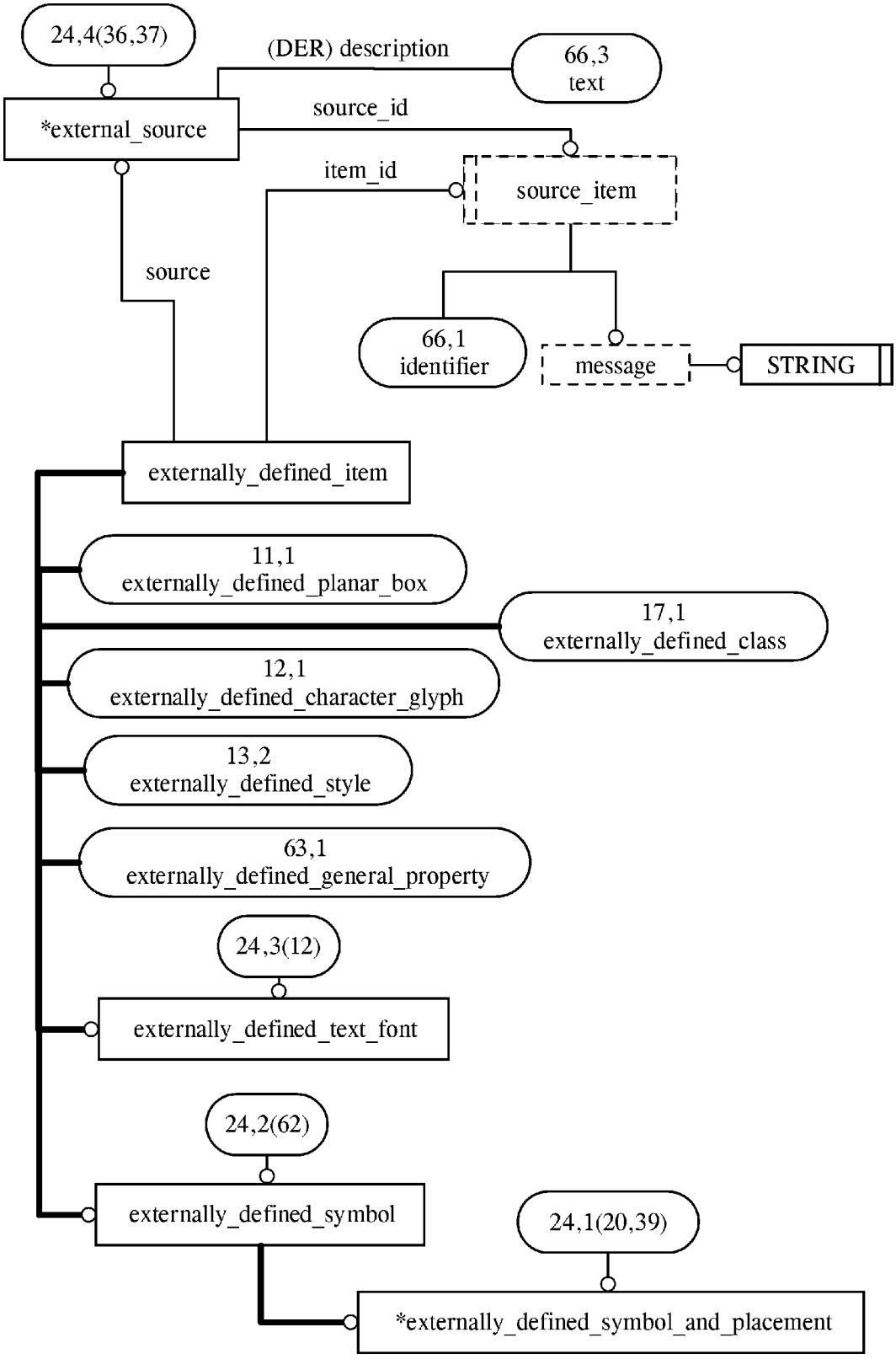

Figure H.24 — AIM EXPRESS-G diagram (24 of 66)

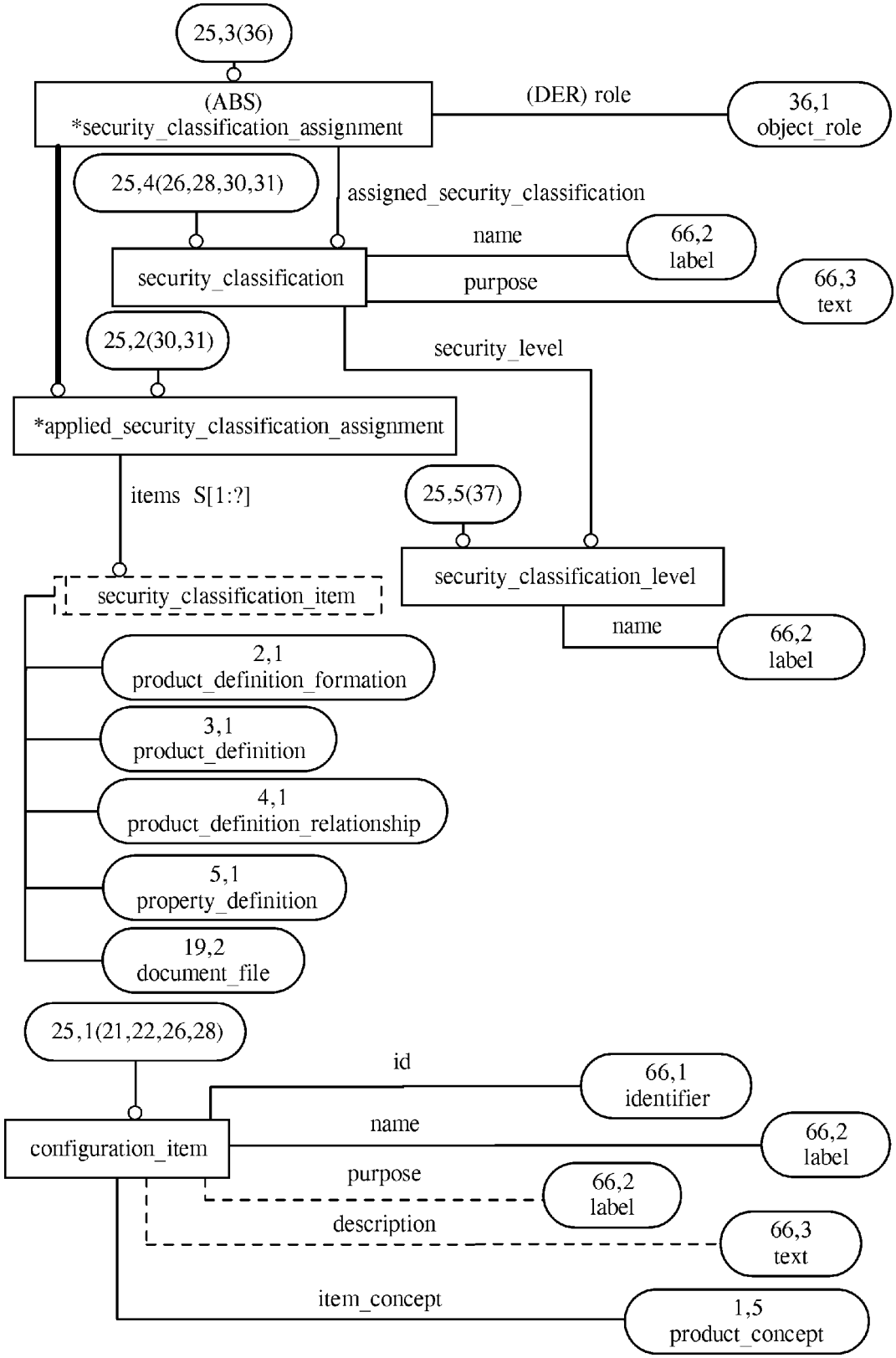

Figure H.25 — AIM EXPRESS-G diagram (25 of 66)

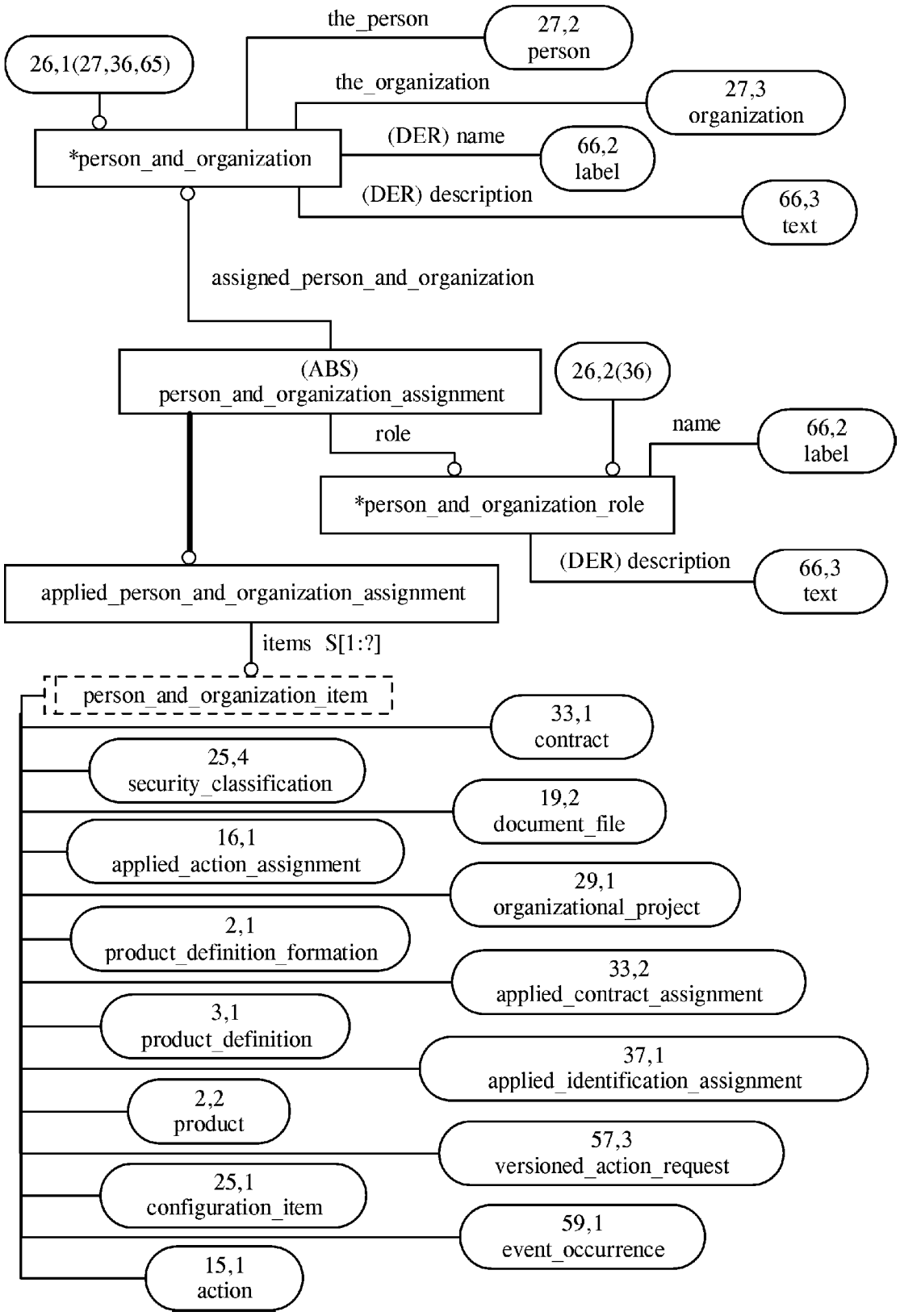

Figure H.26 — AIM EXPRESS-G diagram (26 of 66)

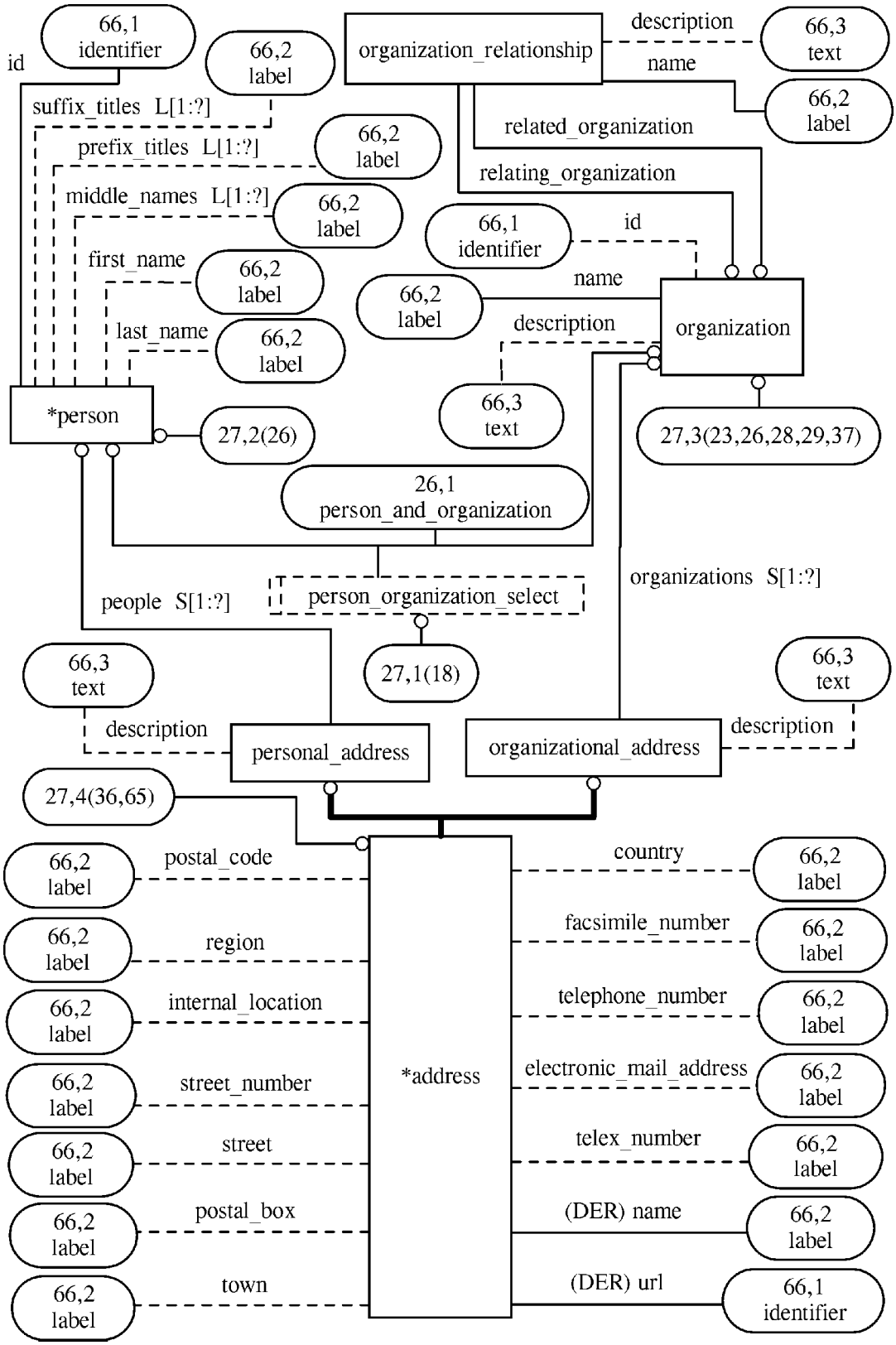

Figure H.27 — AIM EXPRESS-G diagram (27 of 66)

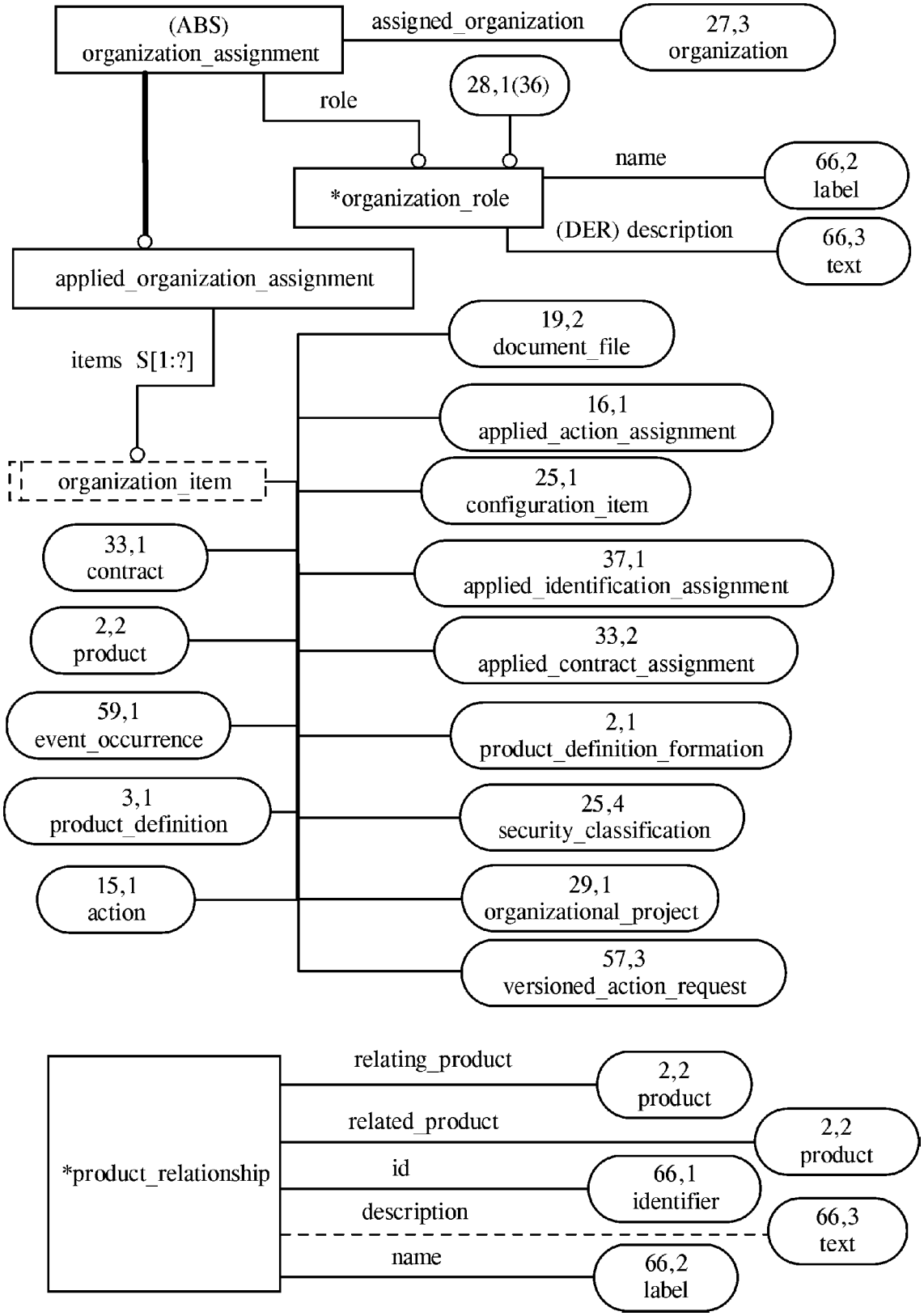

Figure H.28 — AIM EXPRESS-G diagram (28 of 66)

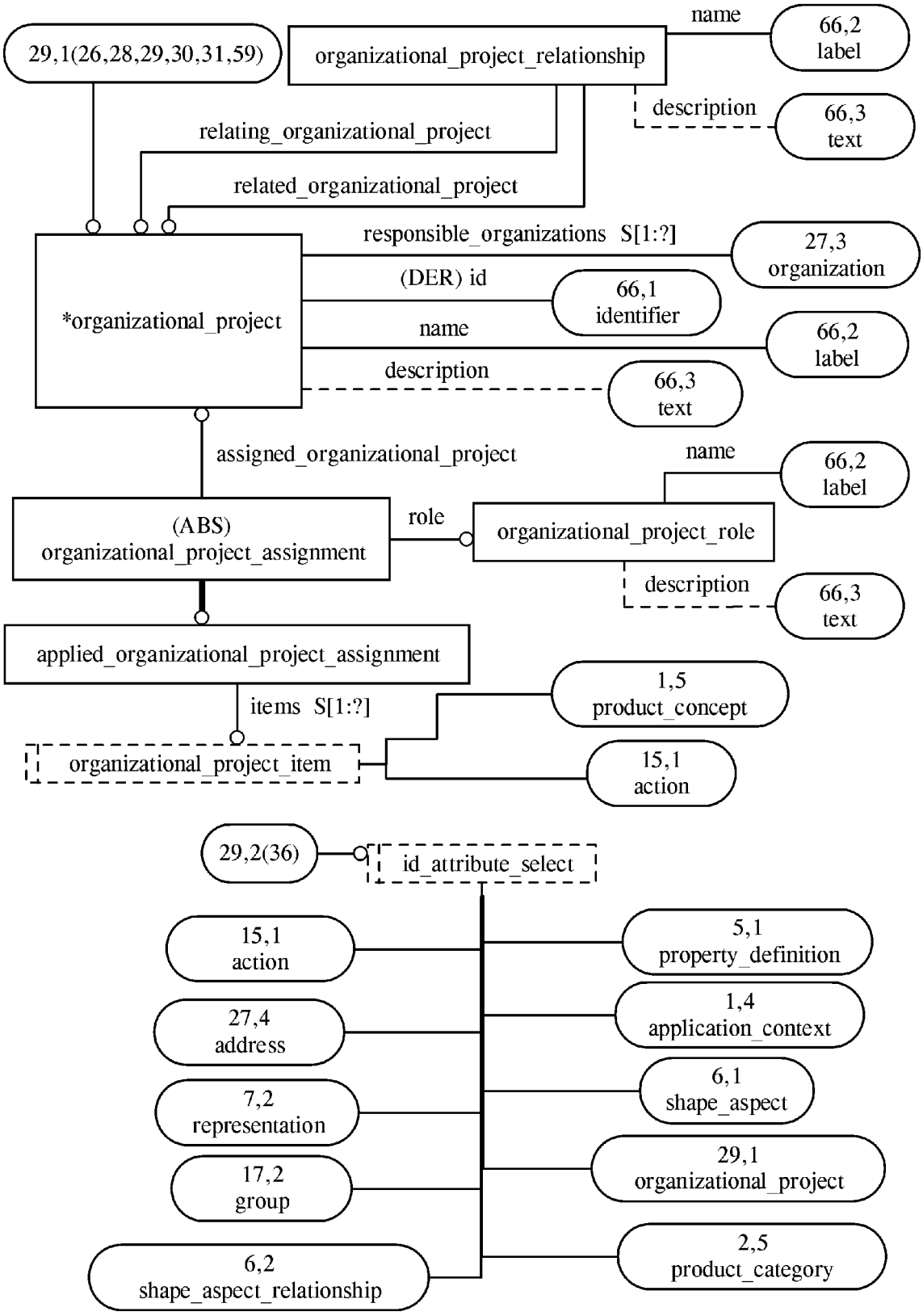

Figure H.29 — AIM EXPRESS-G diagram (29 of 66)

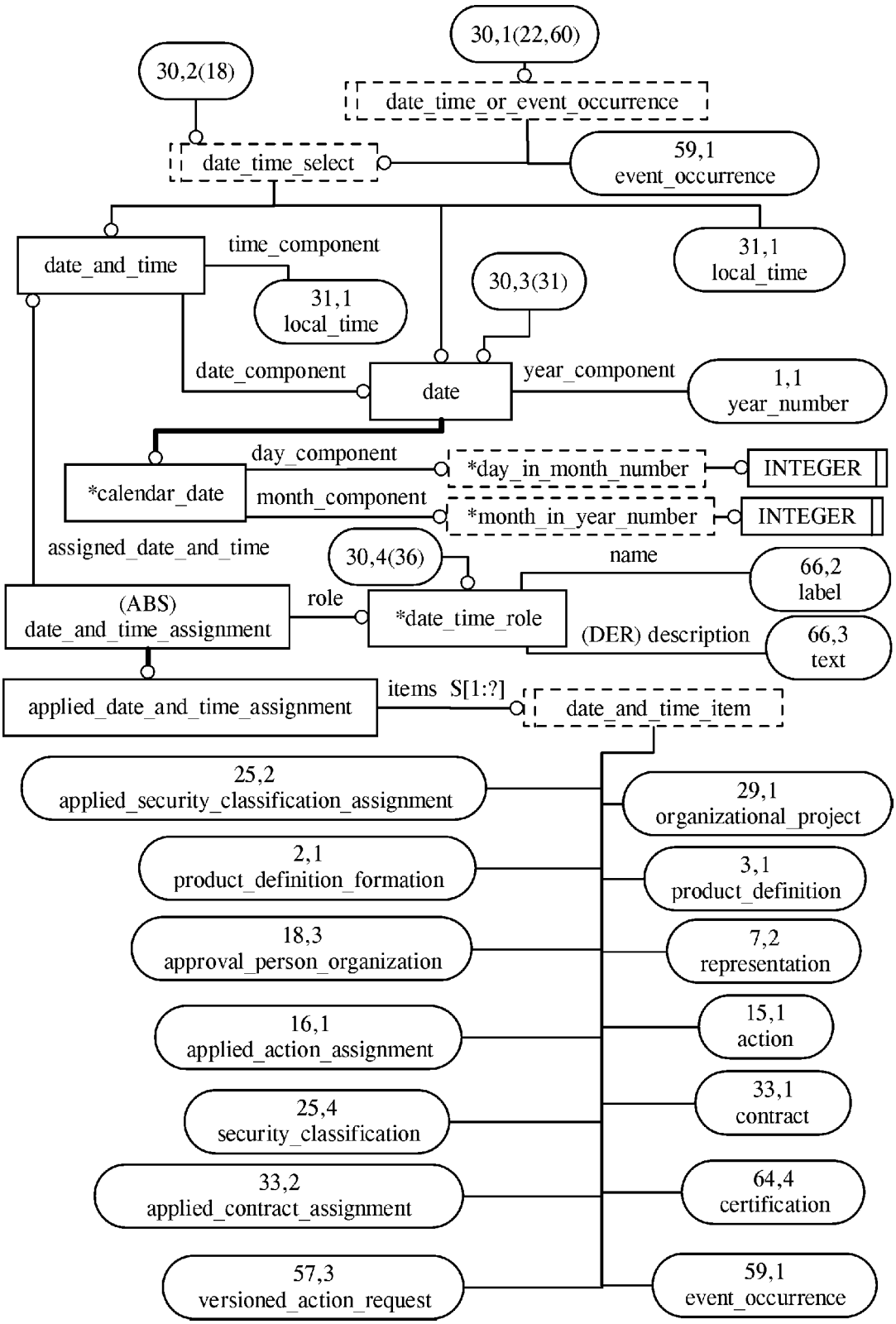

Figure H.30 — AIM EXPRESS-G diagram (30 of 66)

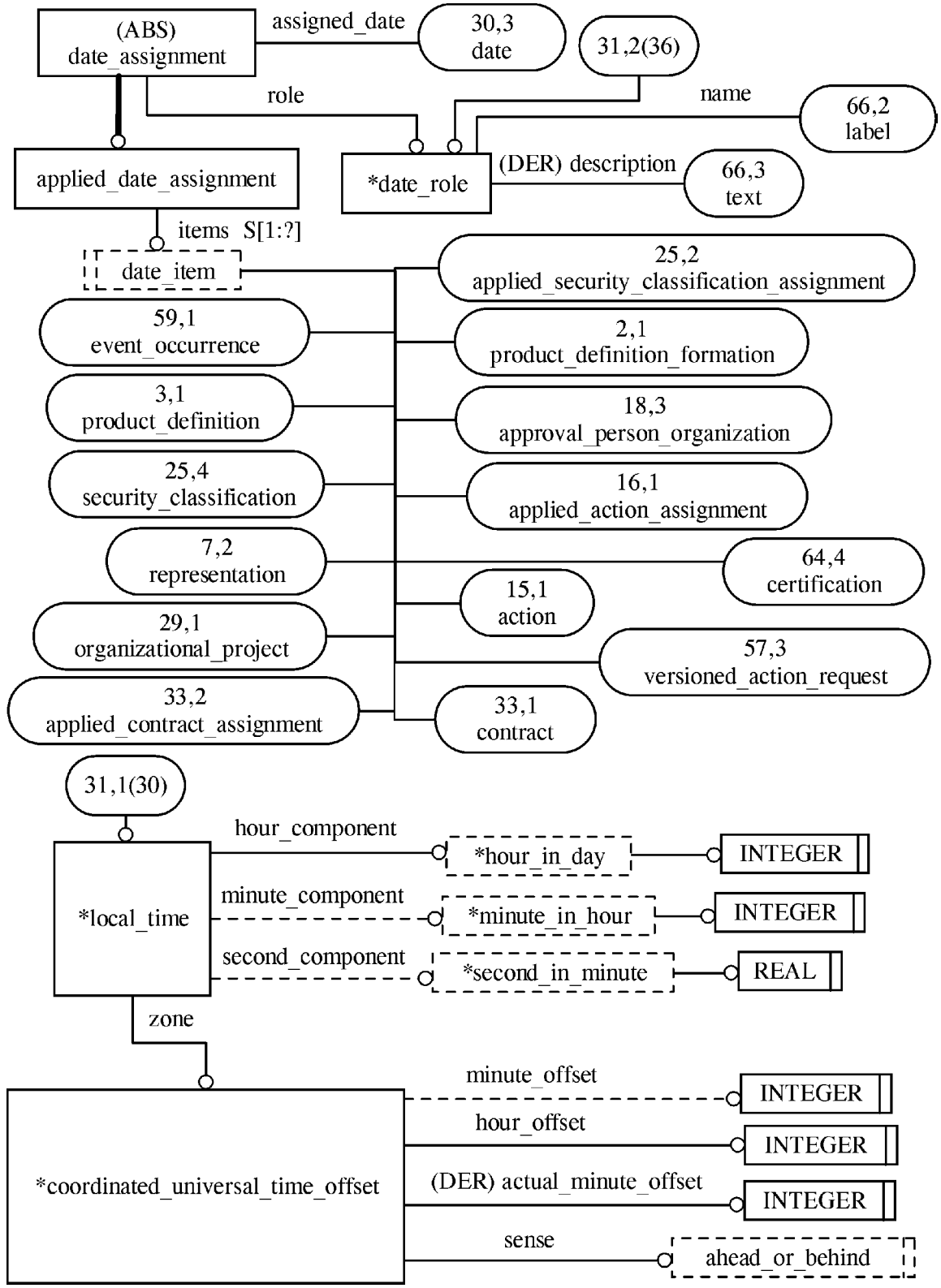

Figure H.31 — AIM EXPRESS-G diagram (31 of 66)

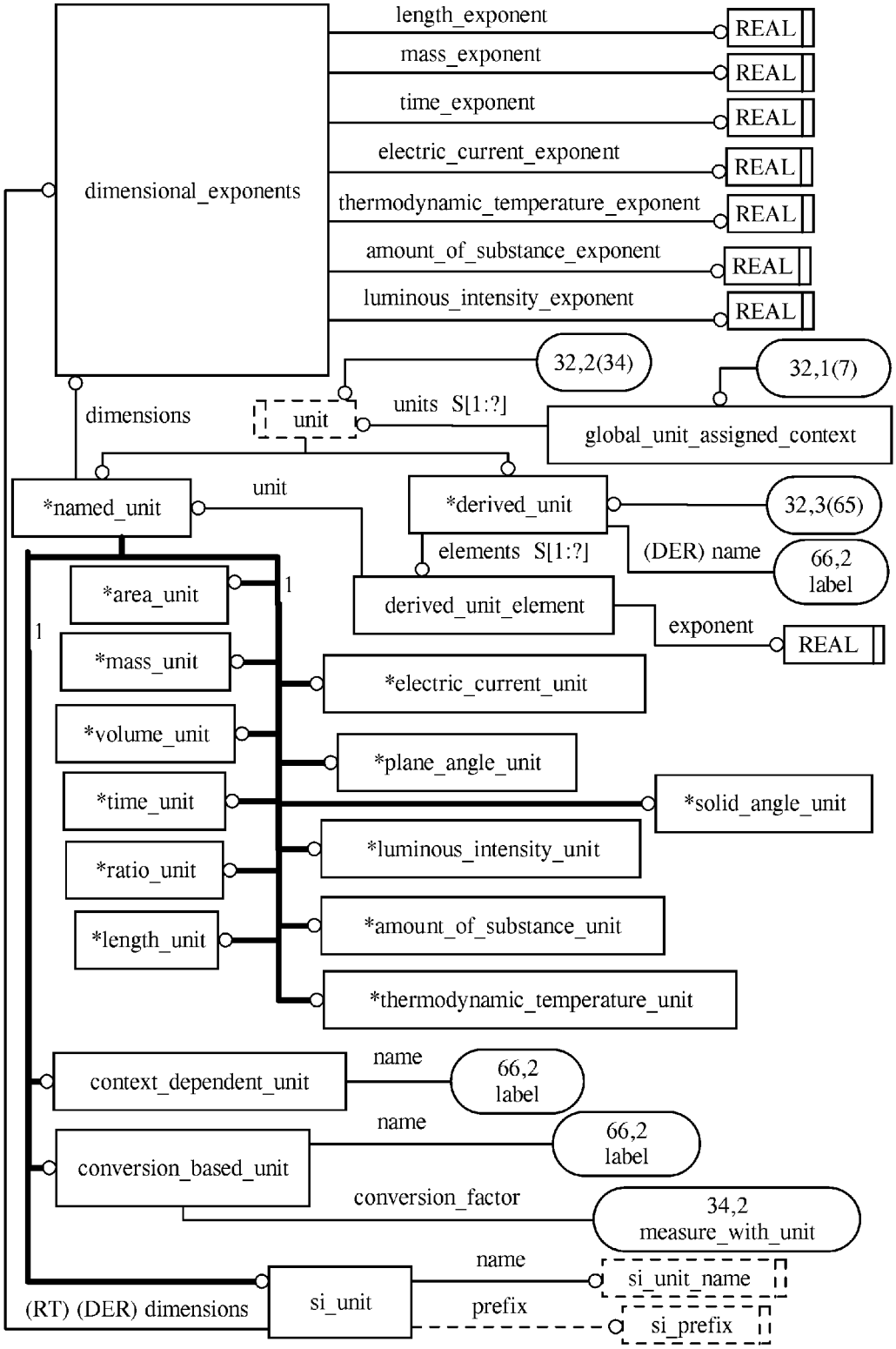

Figure H.32 — AIM EXPRESS-G diagram (32 of 66)

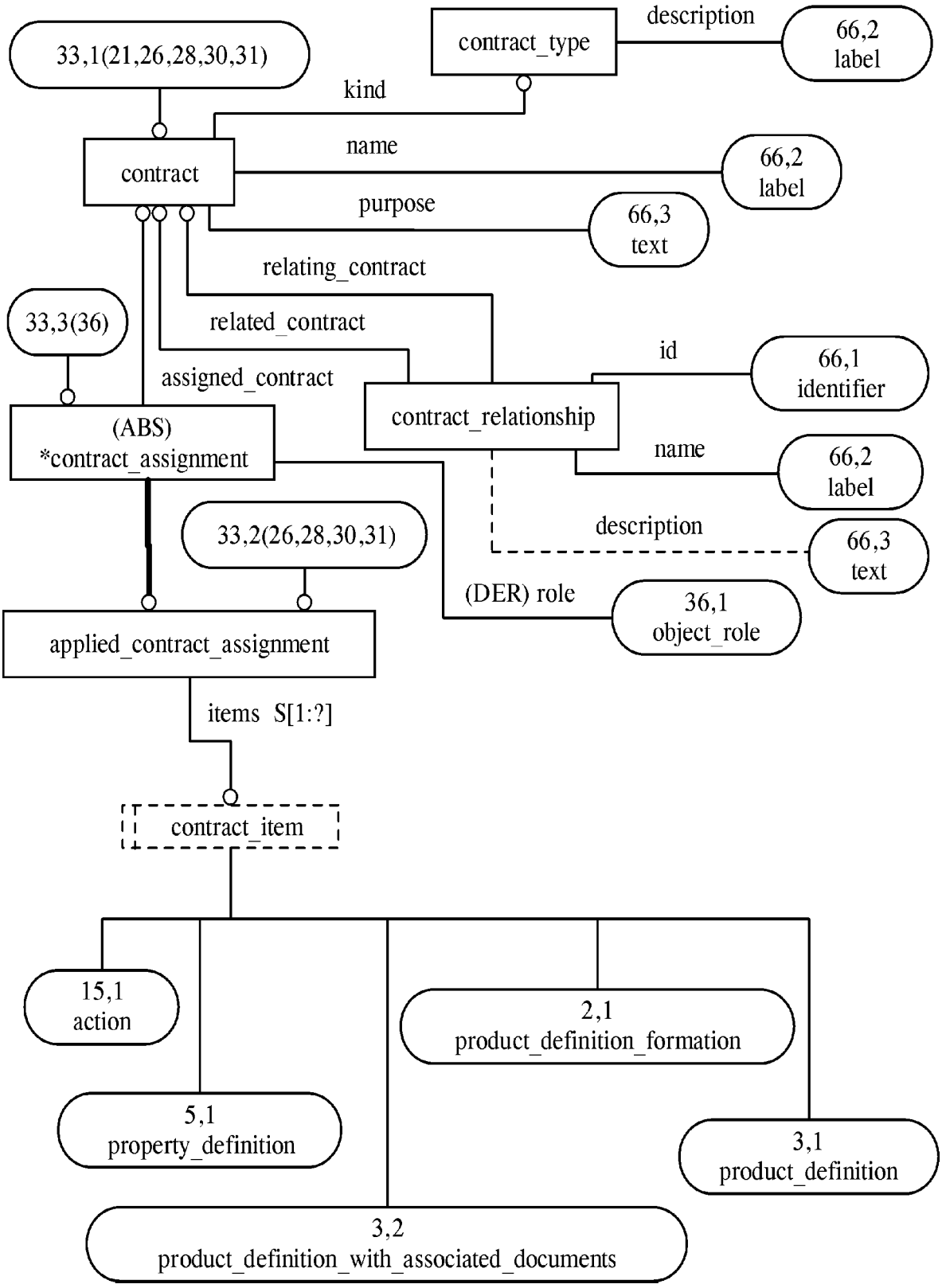

Figure H.33 — AIM EXPRESS-G diagram (33 of 66)

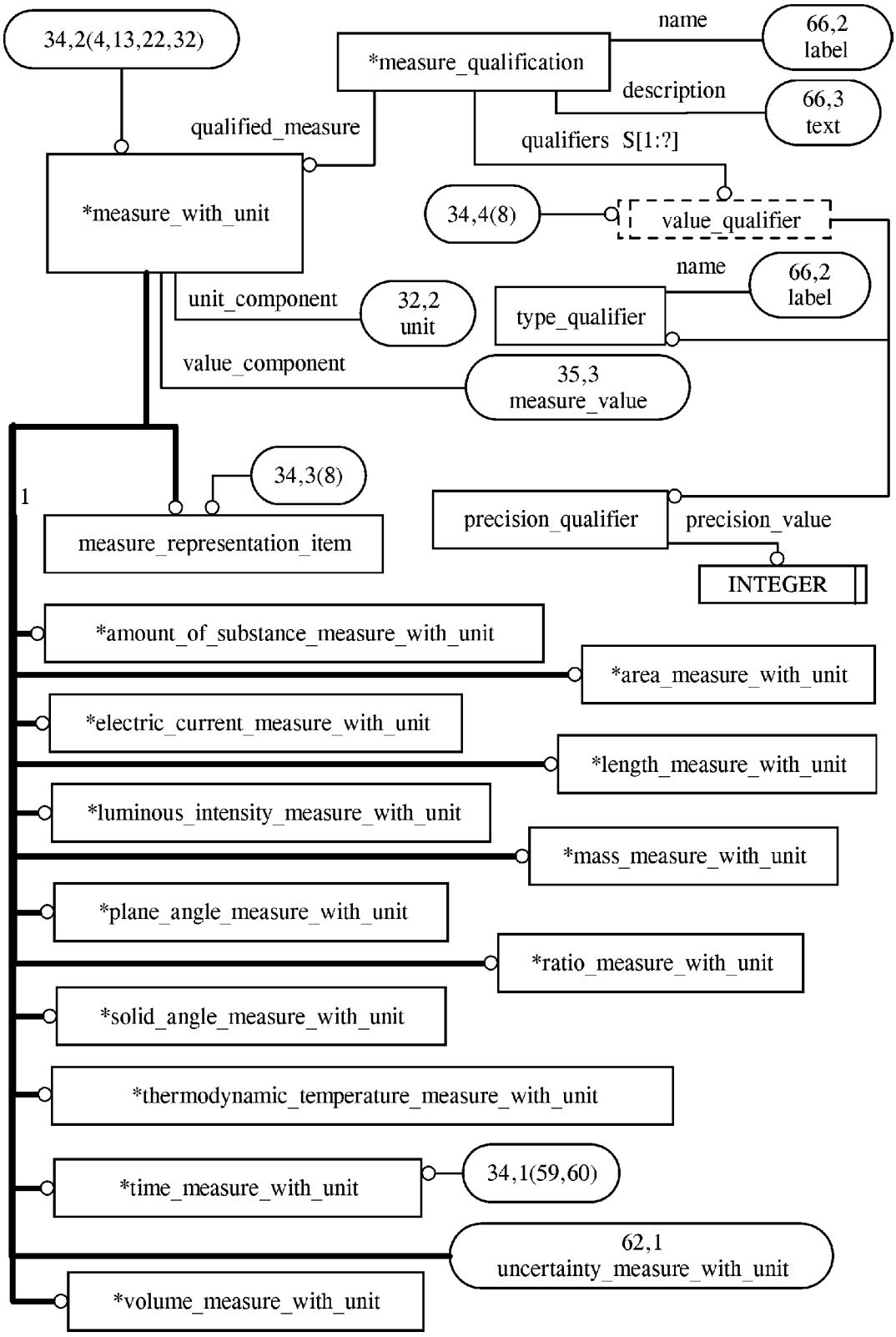

Figure H.34 — AIM EXPRESS-G diagram (34 of 66)

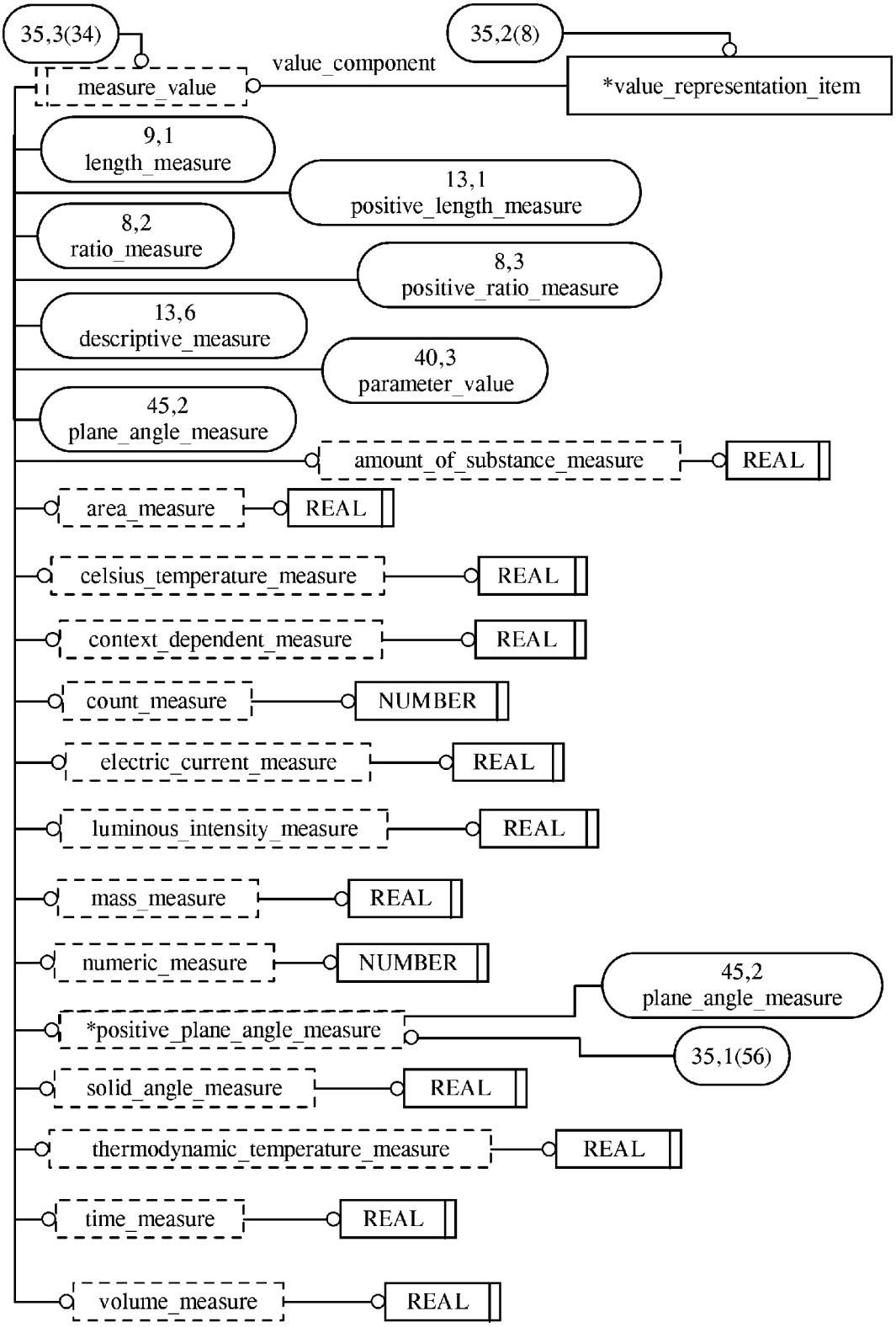

Figure H.35 — AIM EXPRESS-G diagram (35 of 66)

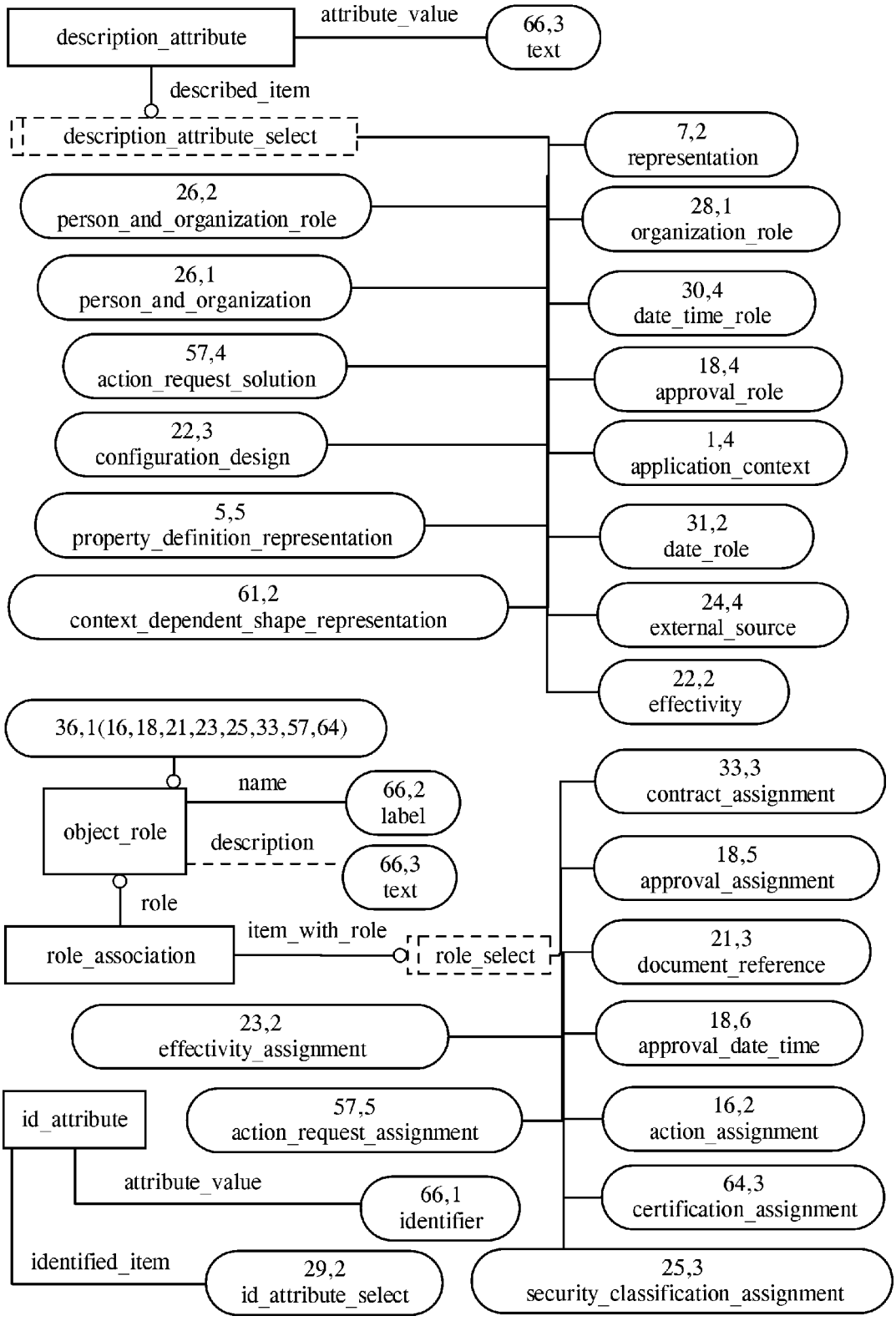

Figure H.36 — AIM EXPRESS-G diagram (36 of 66)

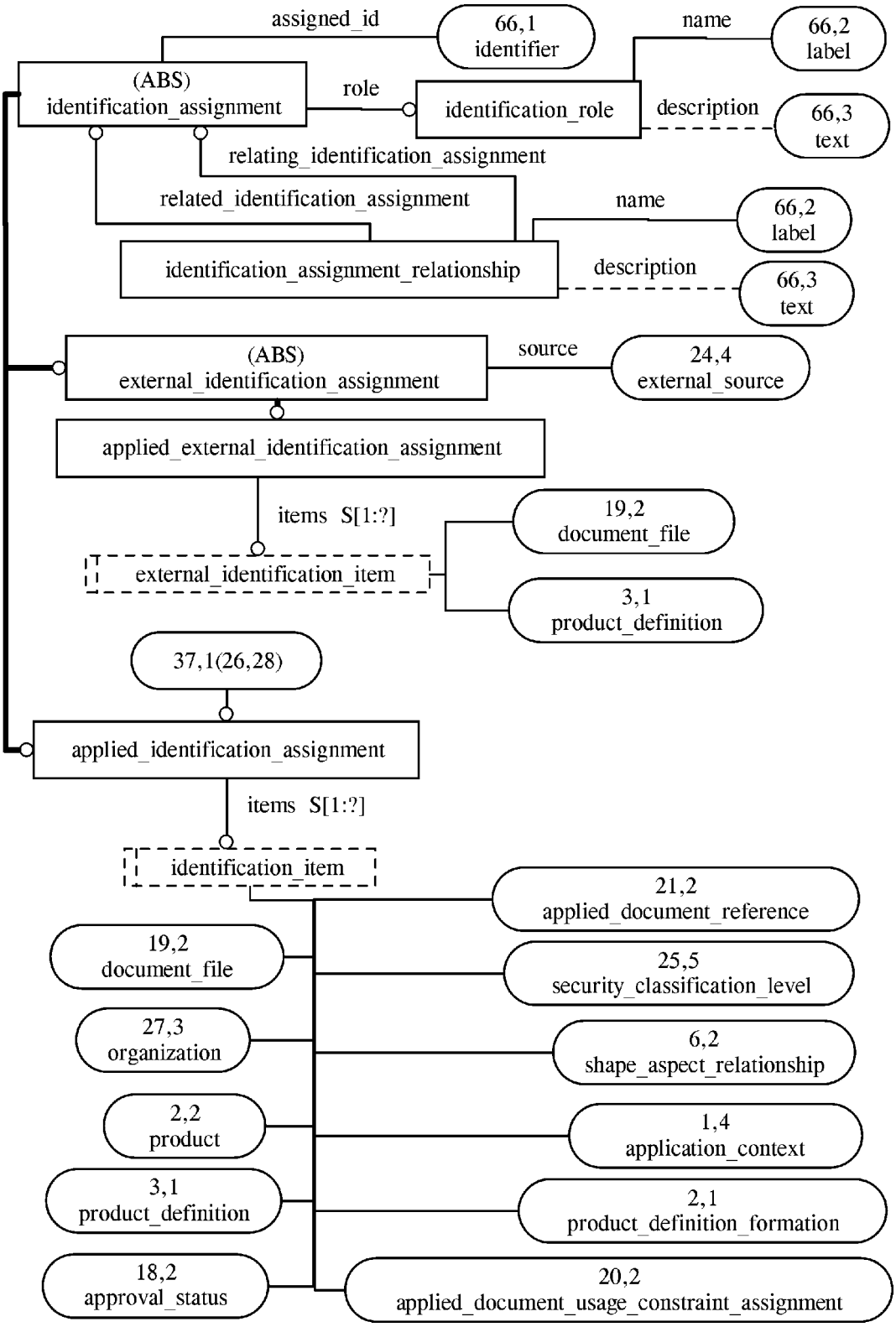

Figure H.37 — AIM EXPRESS-G diagram (37 of 66)

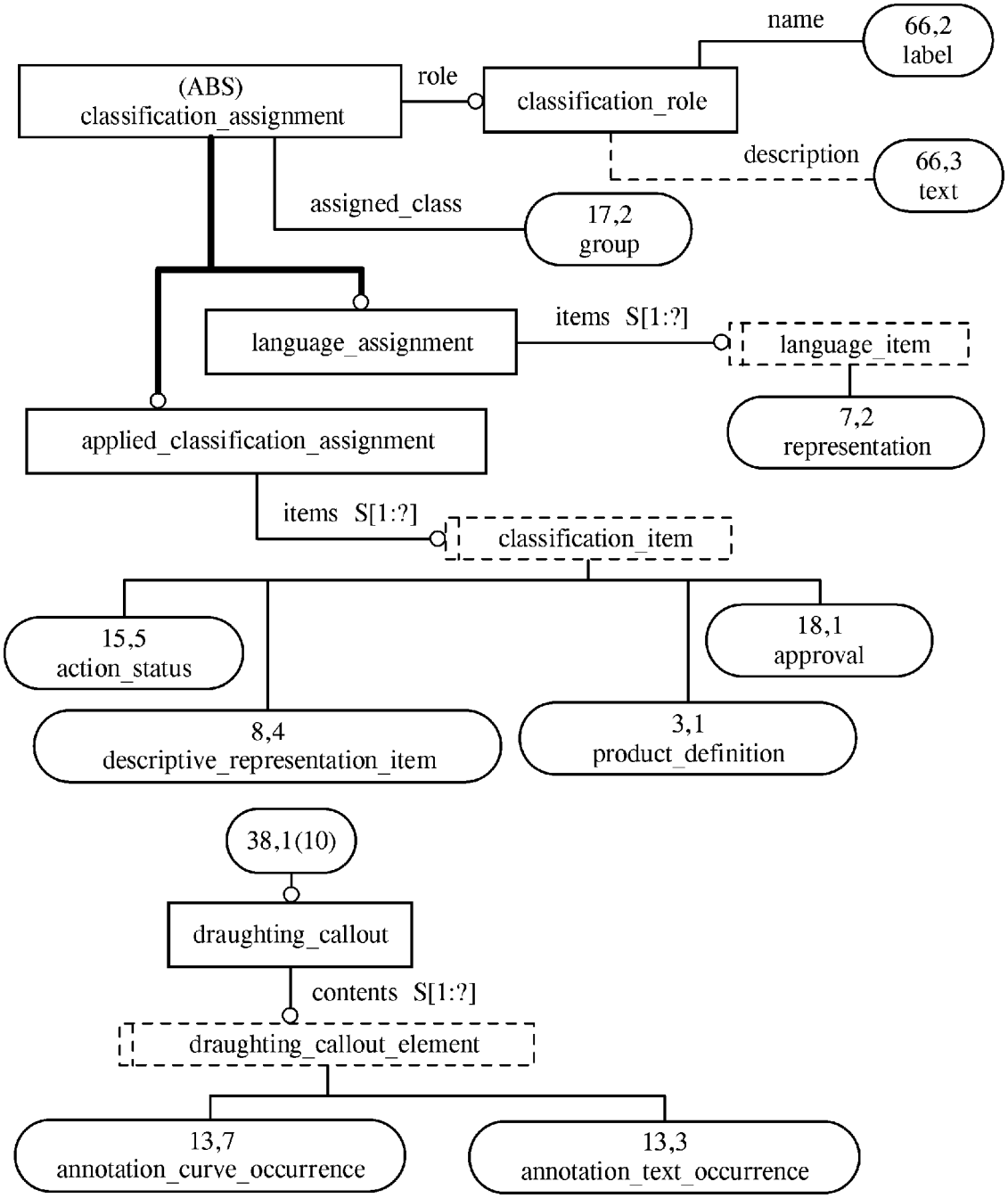

Figure H.38 — AIM EXPRESS-G diagram (38 of 66)

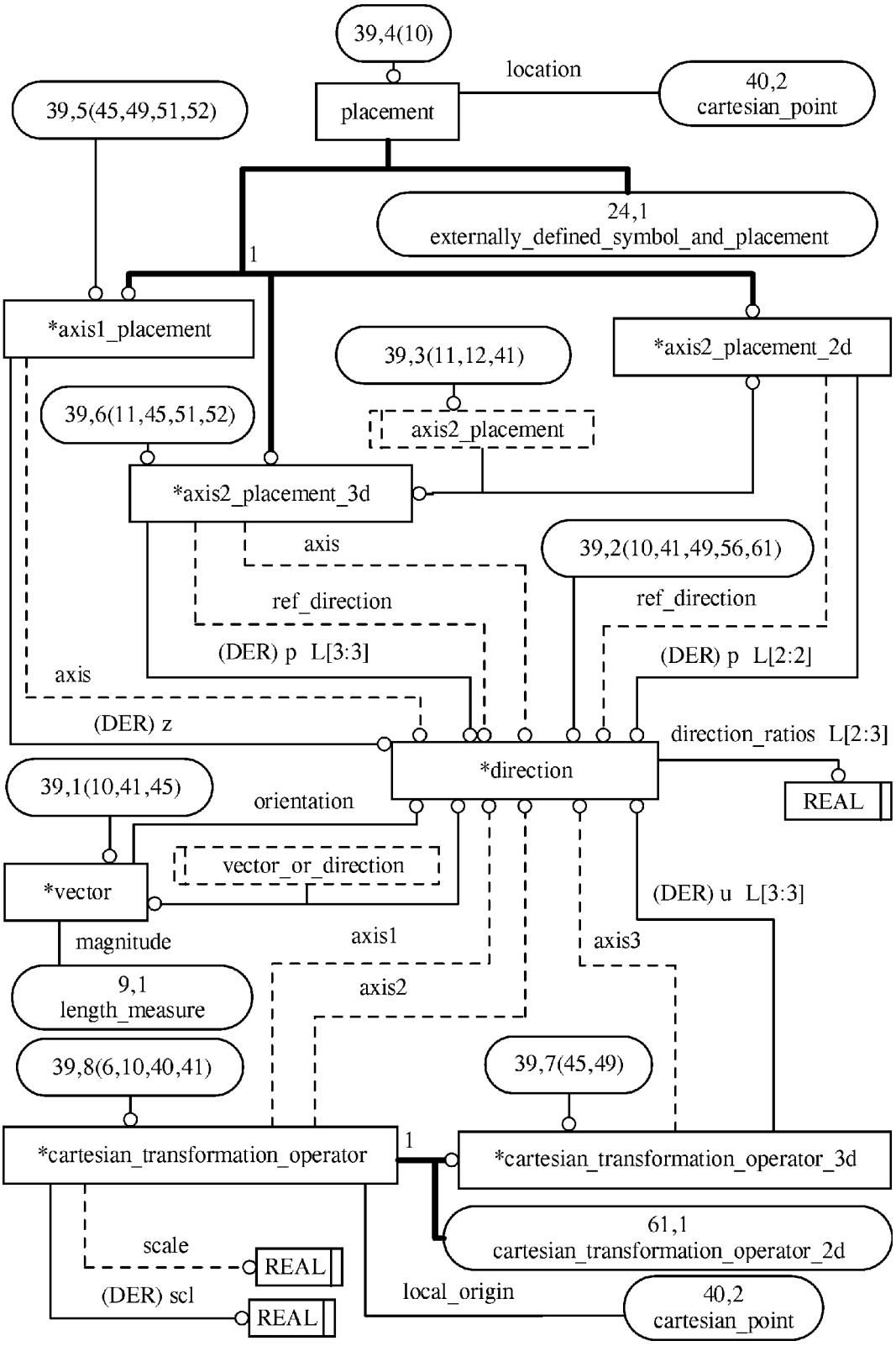

Figure H.39 — AIM EXPRESS-G diagram (39 of 66)

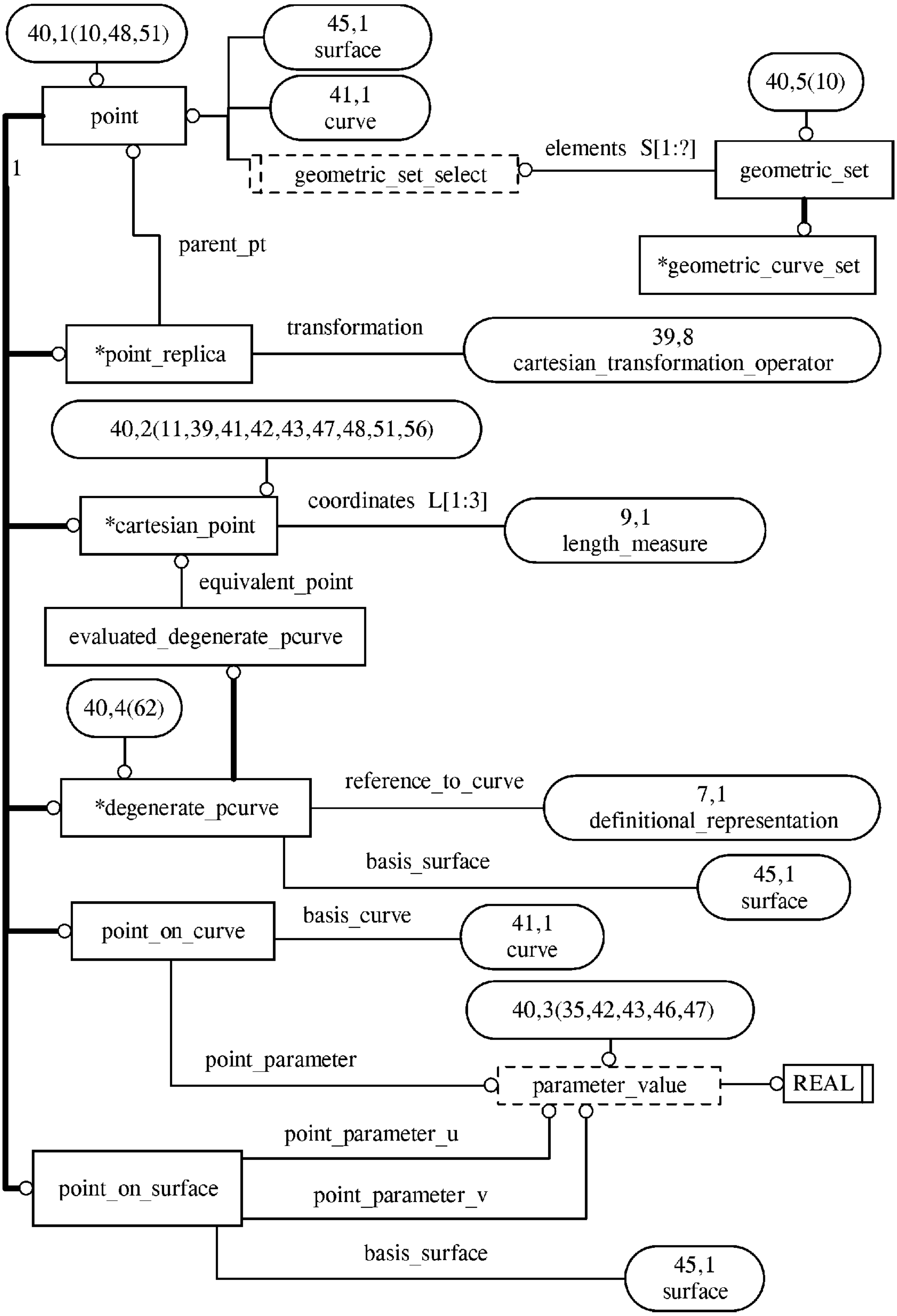

Figure H.40 — AIM EXPRESS-G diagram (40 of 66)

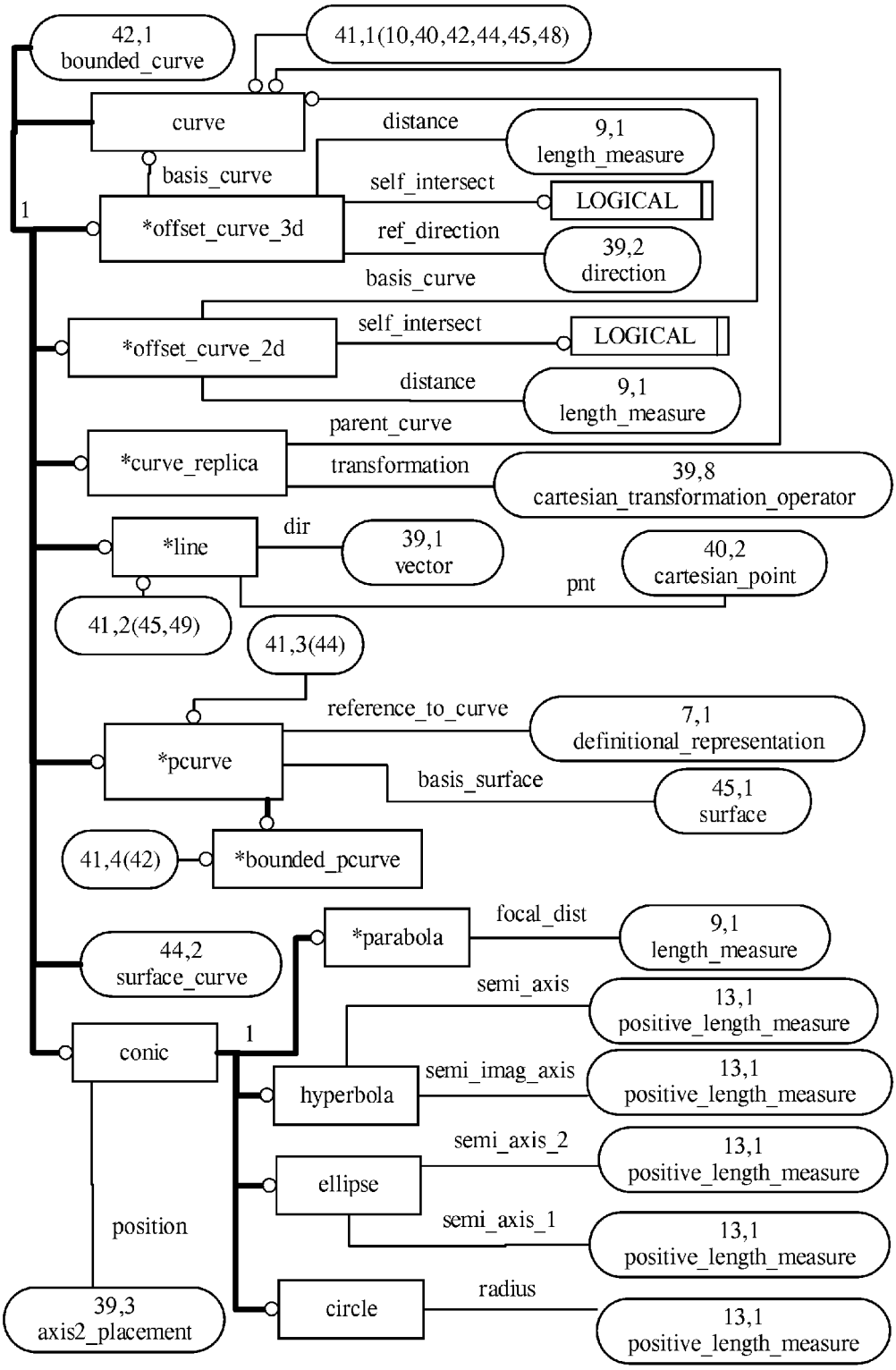

Figure H.41 — AIM EXPRESS-G diagram (41 of 66)

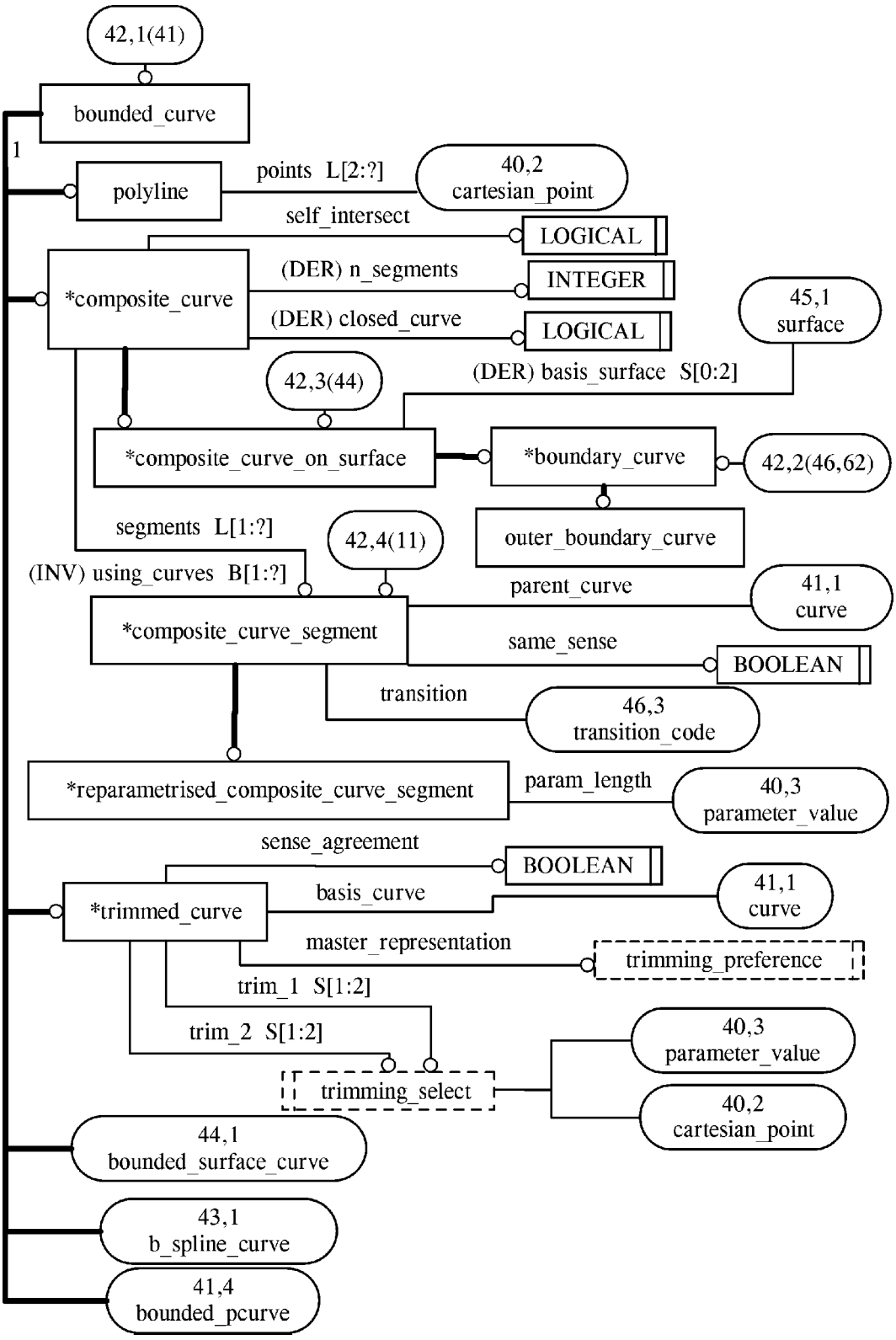

Figure H.42 — AIM EXPRESS-G diagram (42 of 66)

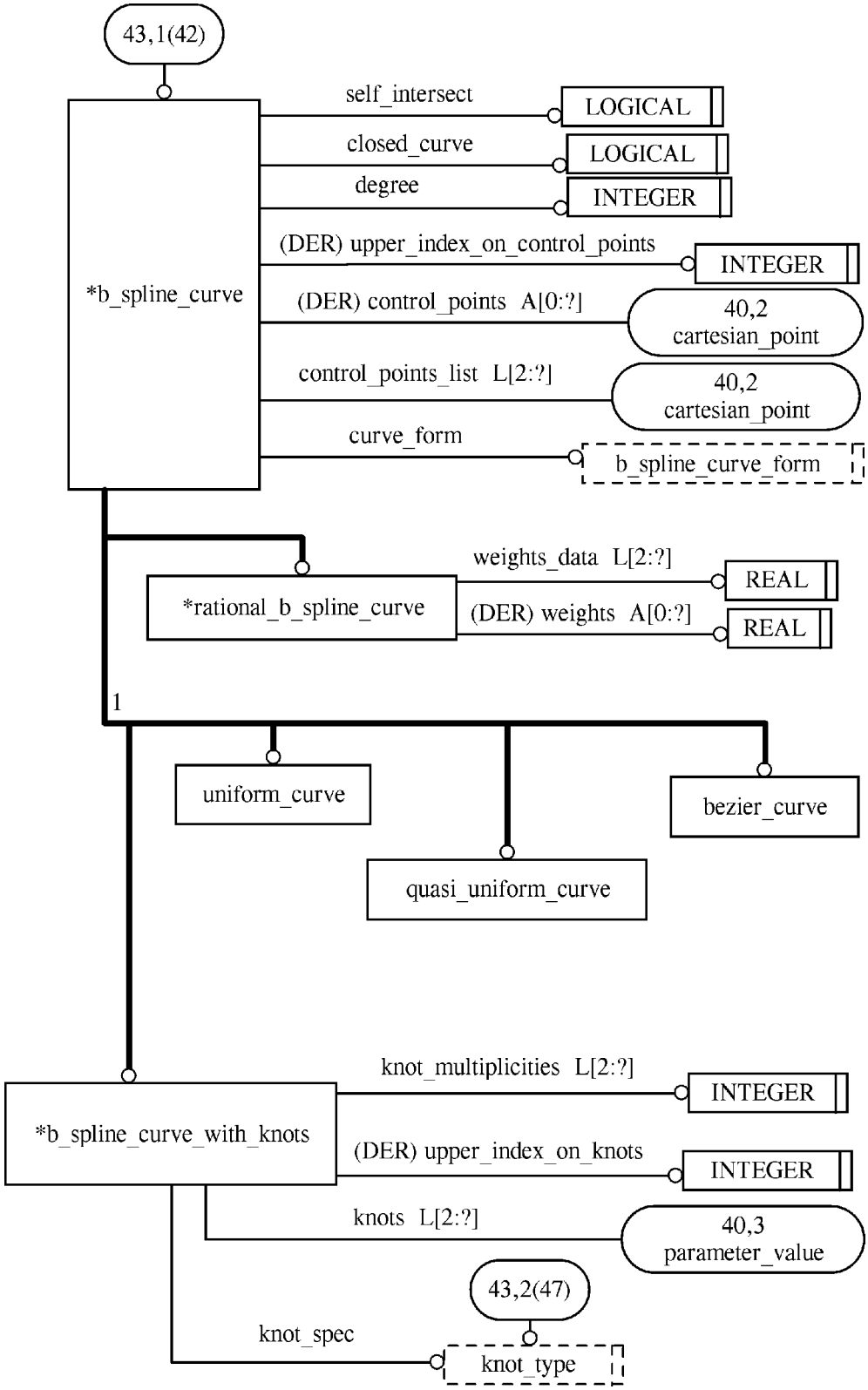

Figure H.43 — AIM EXPRESS-G diagram (43 of 66)

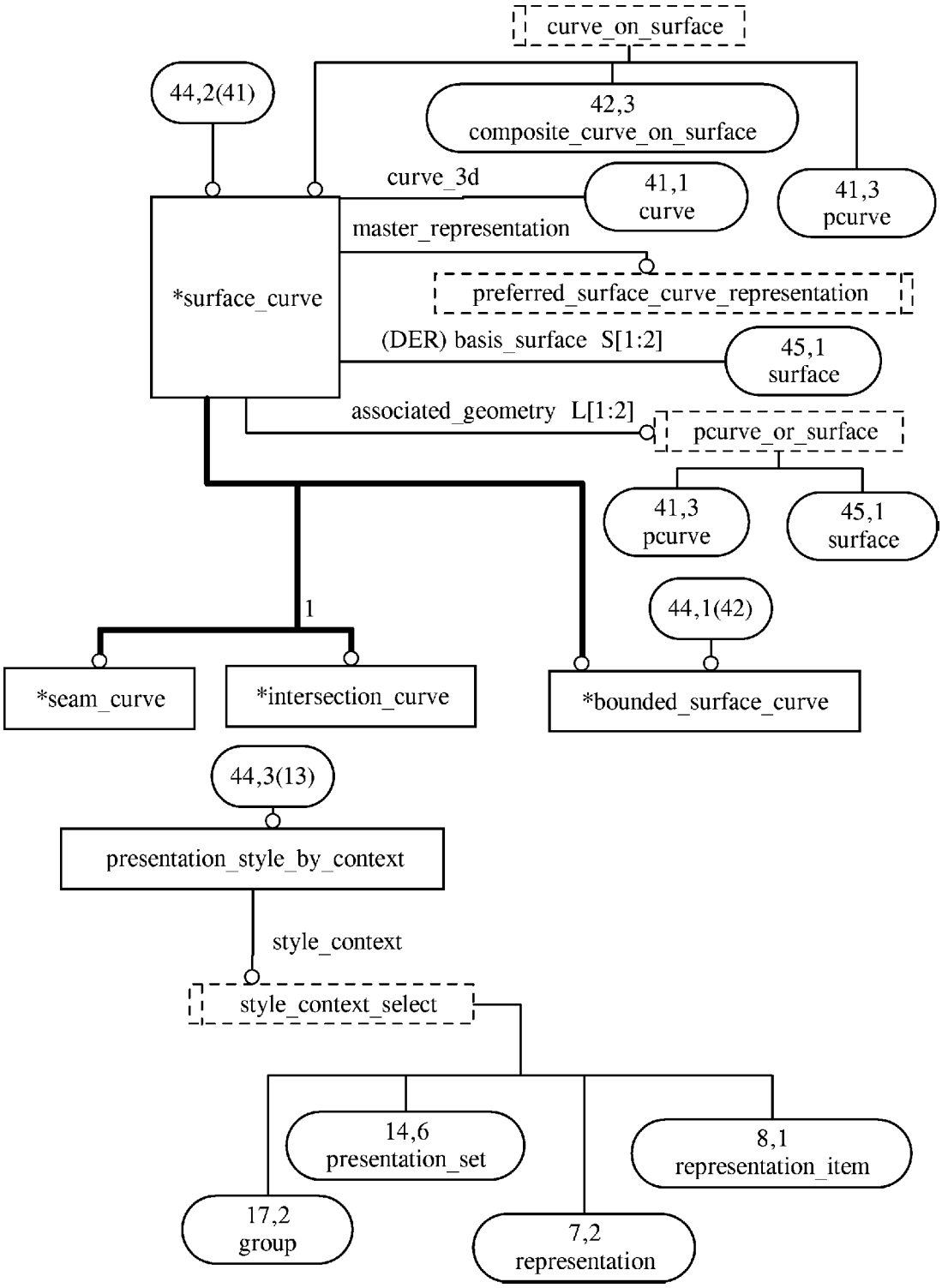

Figure H.44 — AIM EXPRESS-G diagram (44 of 66)

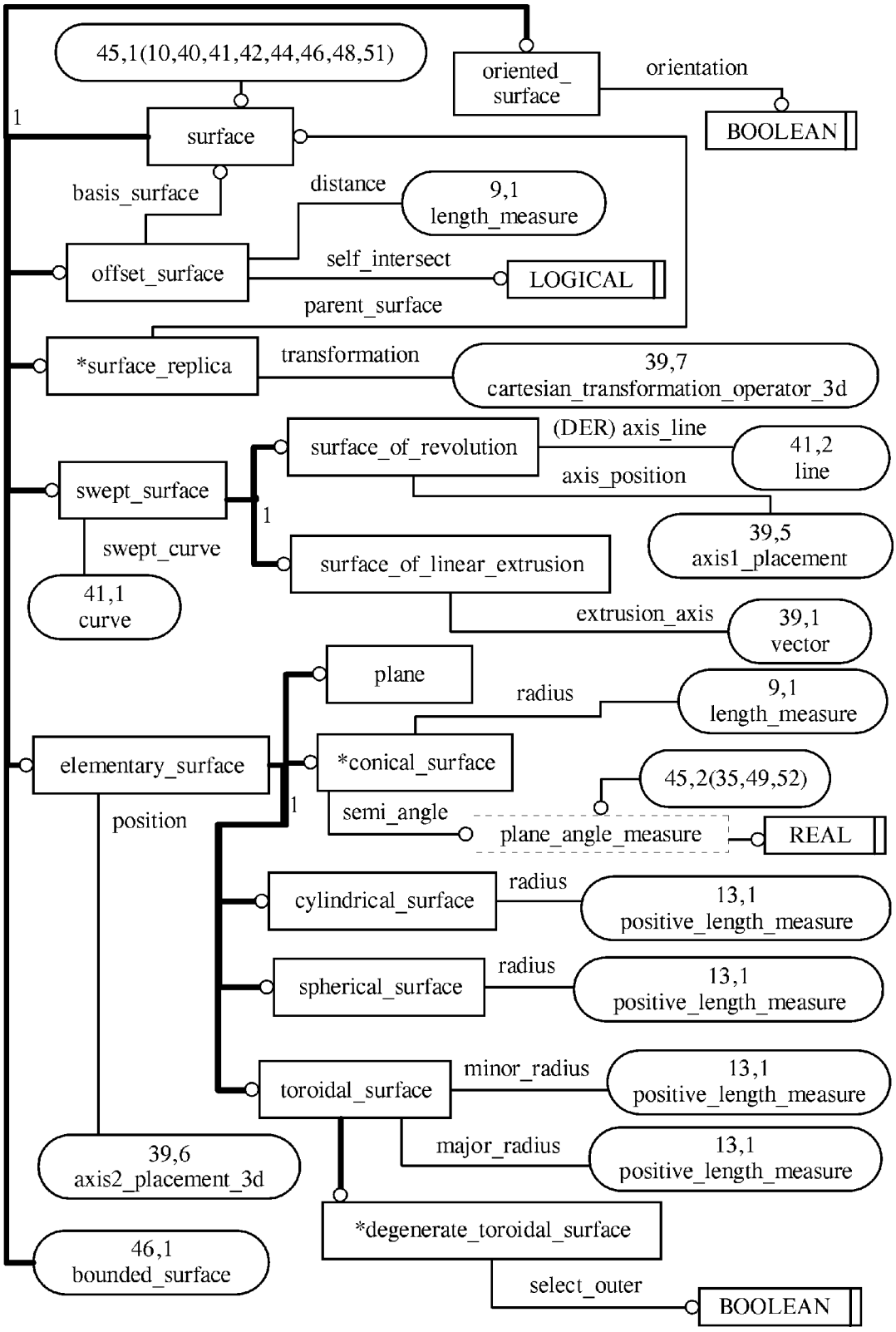

Figure H.45 — AIM EXPRESS-G diagram (45 of 66)

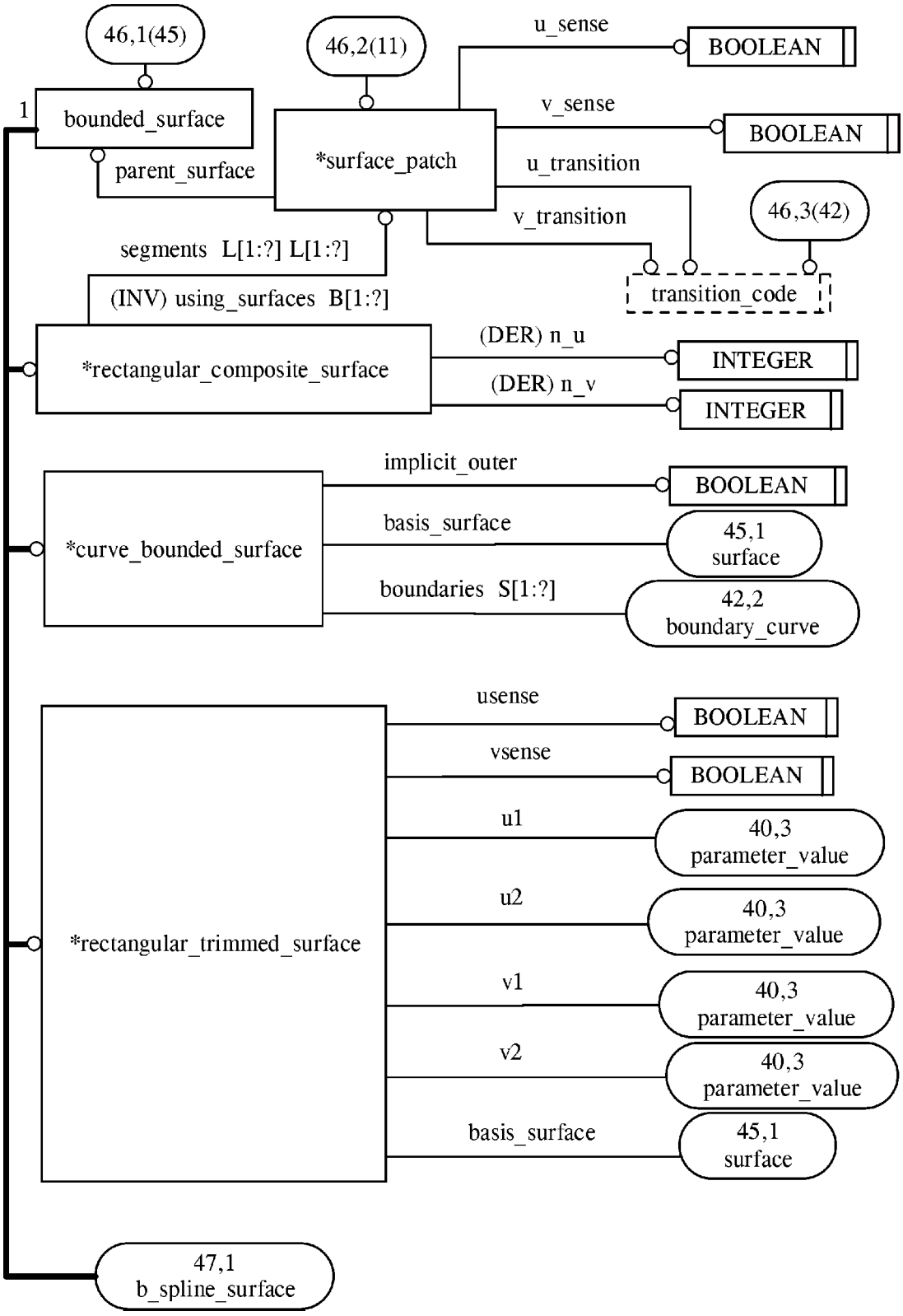

Figure H.46 — AIM EXPRESS-G diagram (46 of 66)

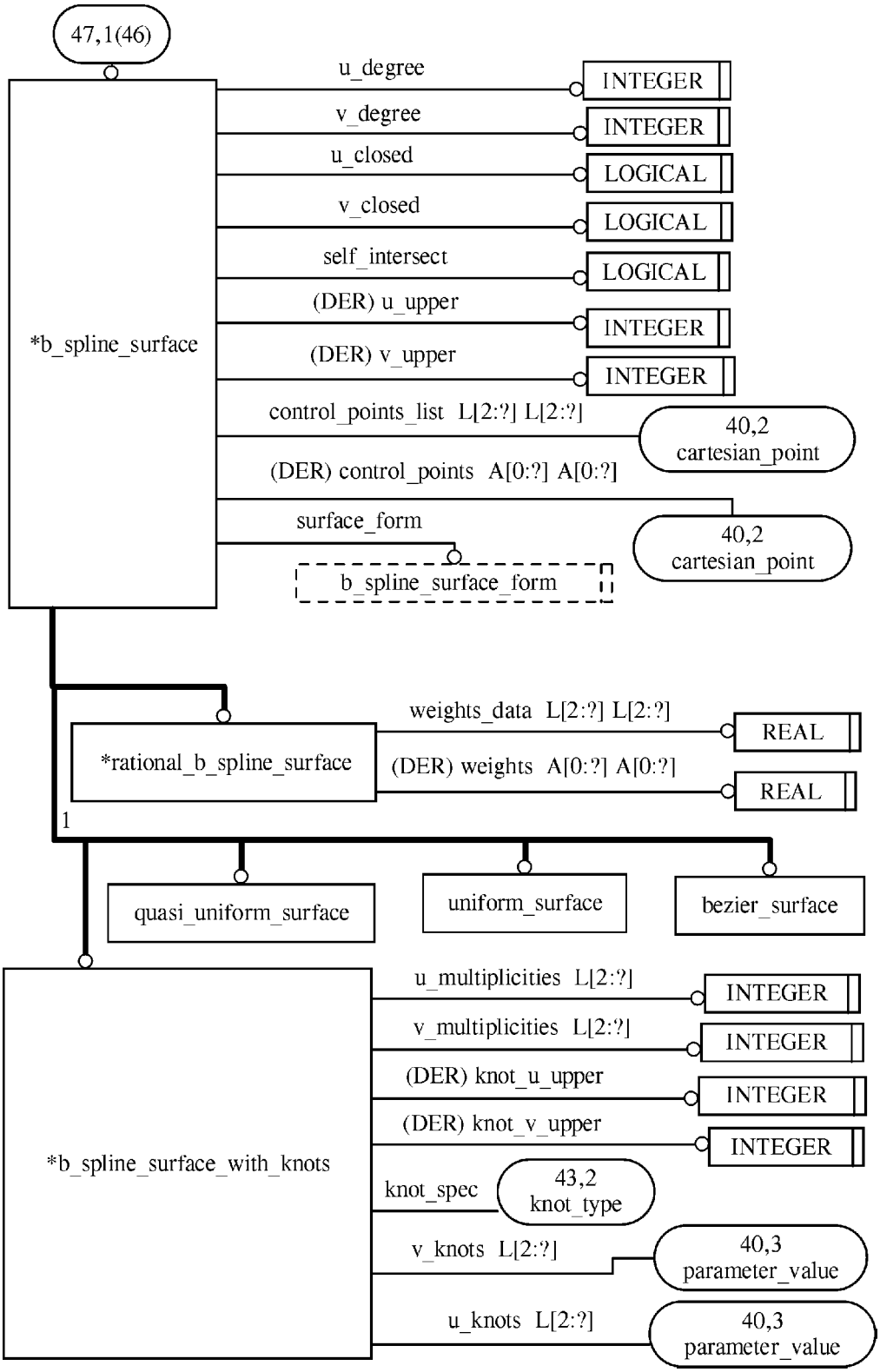

Figure H.47 — AIM EXPRESS-G diagram (47 of 66)

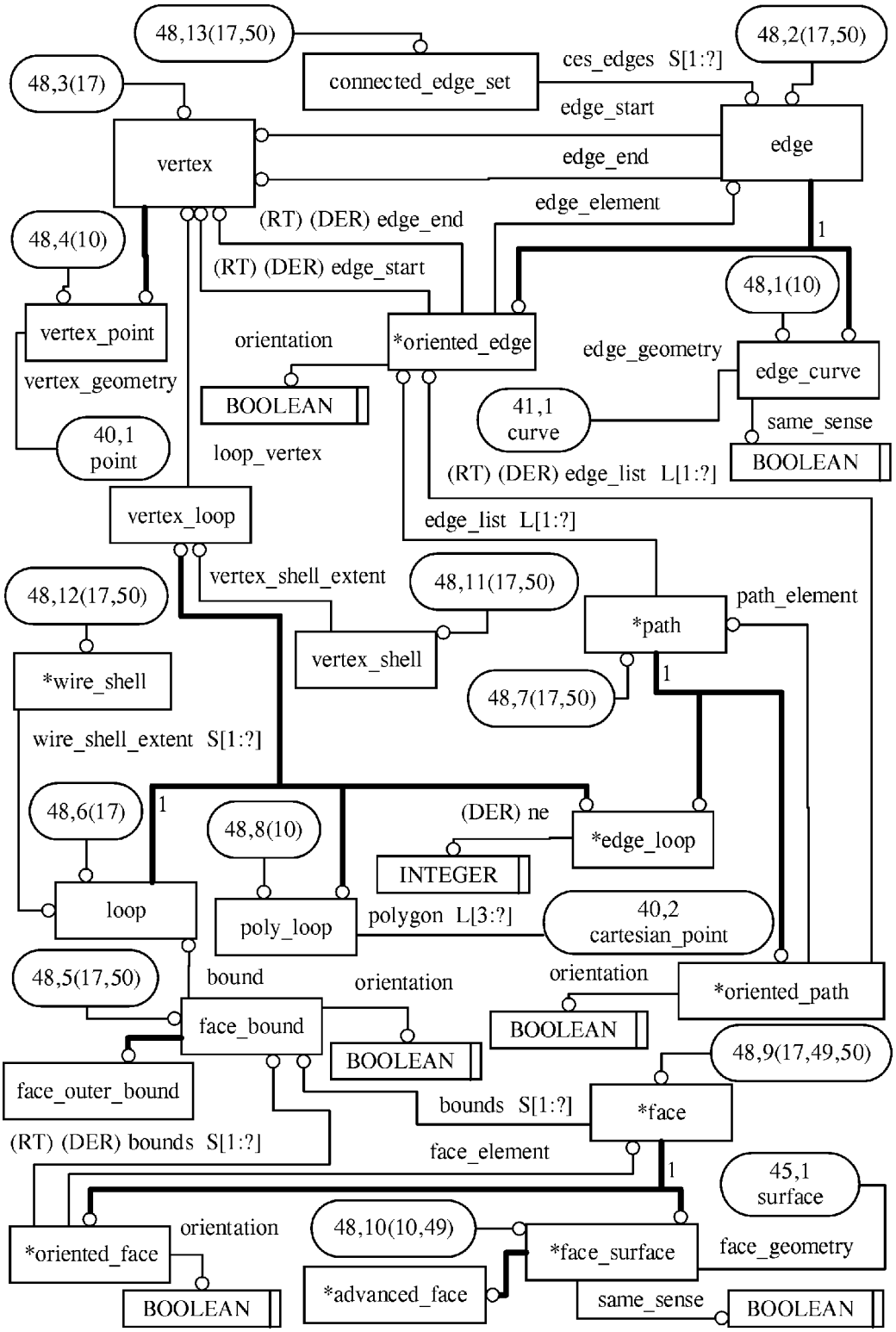

Figure H.48 — AIM EXPRESS-G diagram (48 of 66)

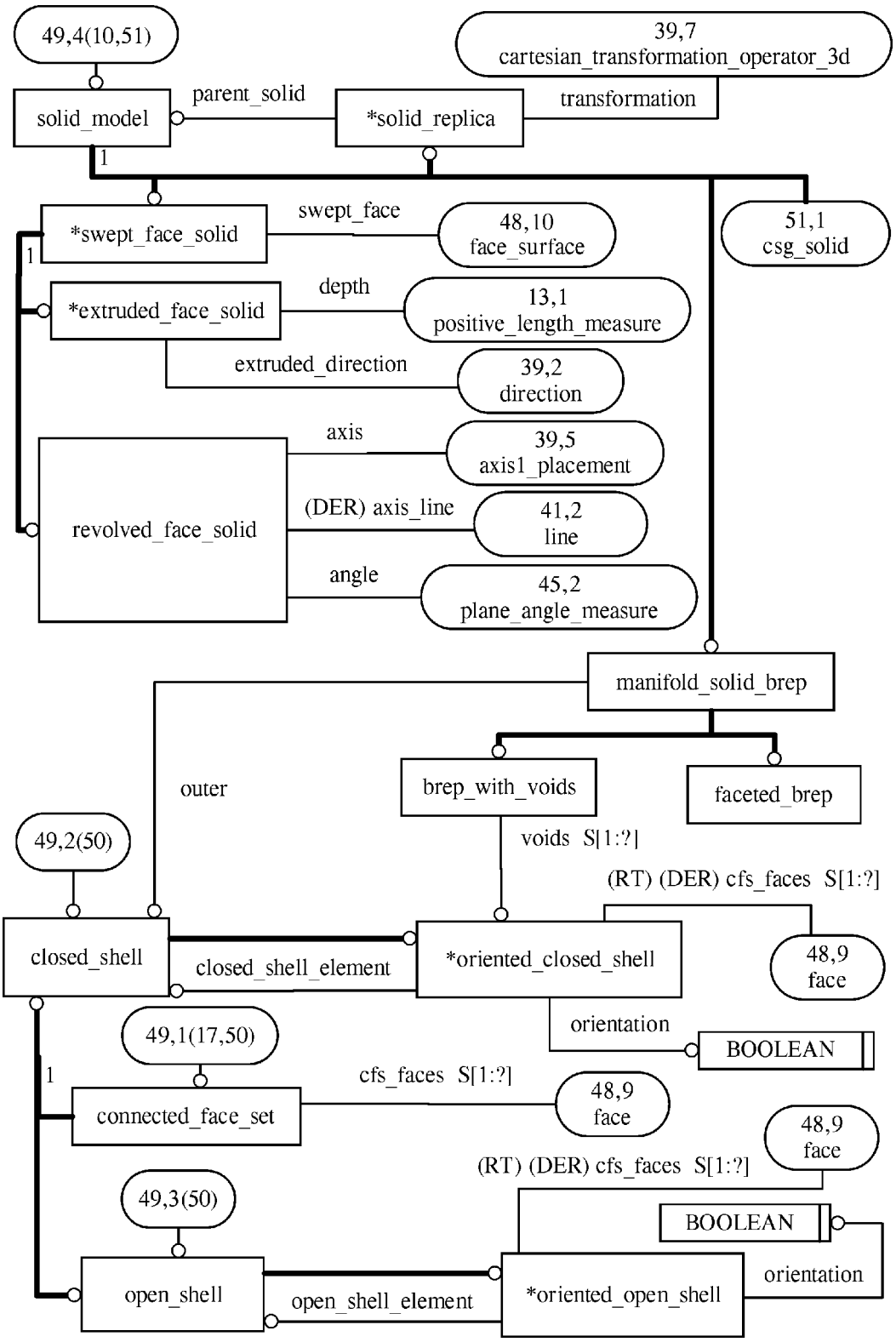

Figure H.49 — AIM EXPRESS-G diagram (49 of 66)

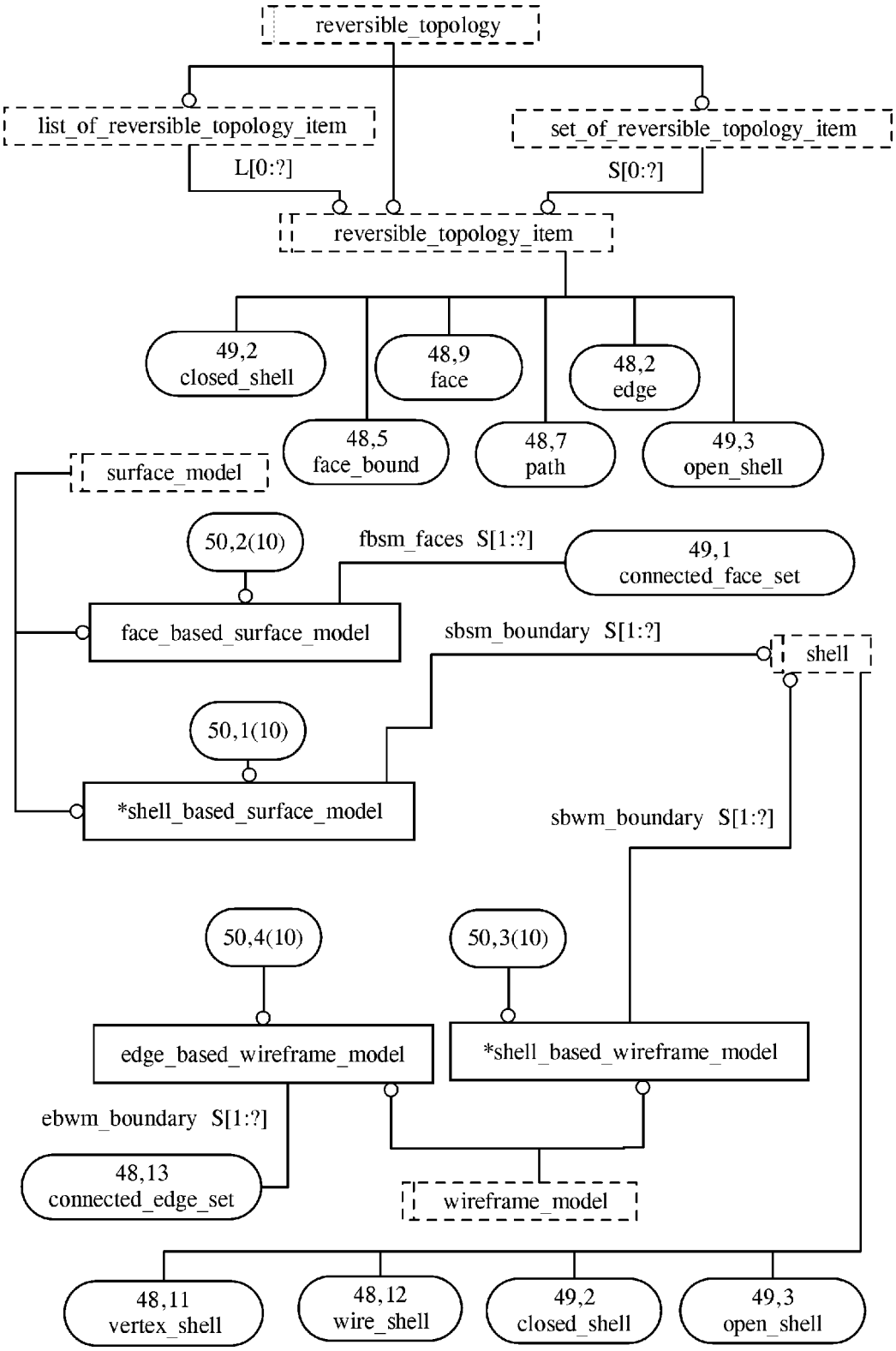

Figure H.50 — AIM EXPRESS-G diagram (50 of 66)

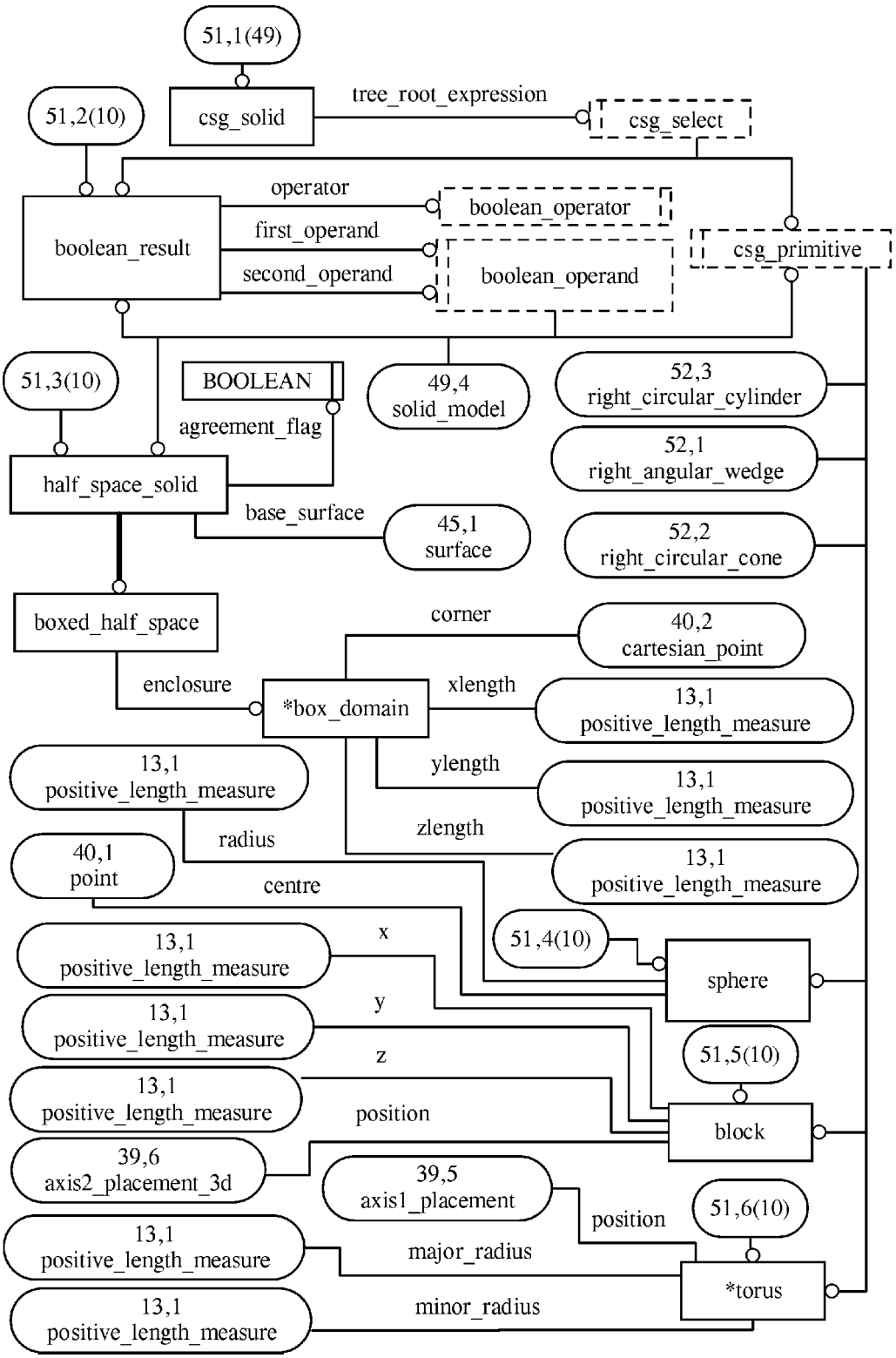

Figure H.51 — AIM EXPRESS-G diagram (51 of 66)

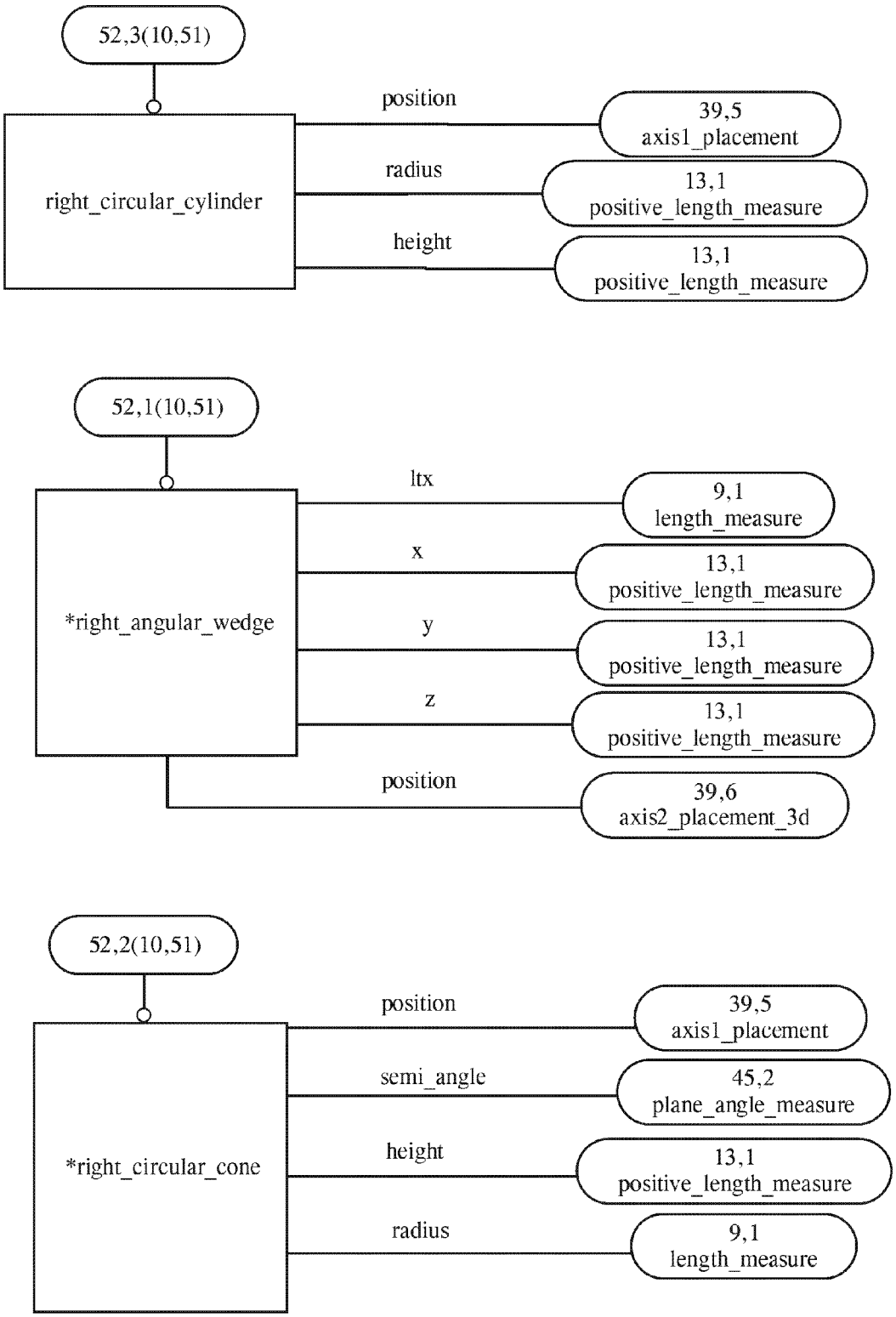

Figure H.52 — AIM EXPRESS-G diagram (52 of 66)

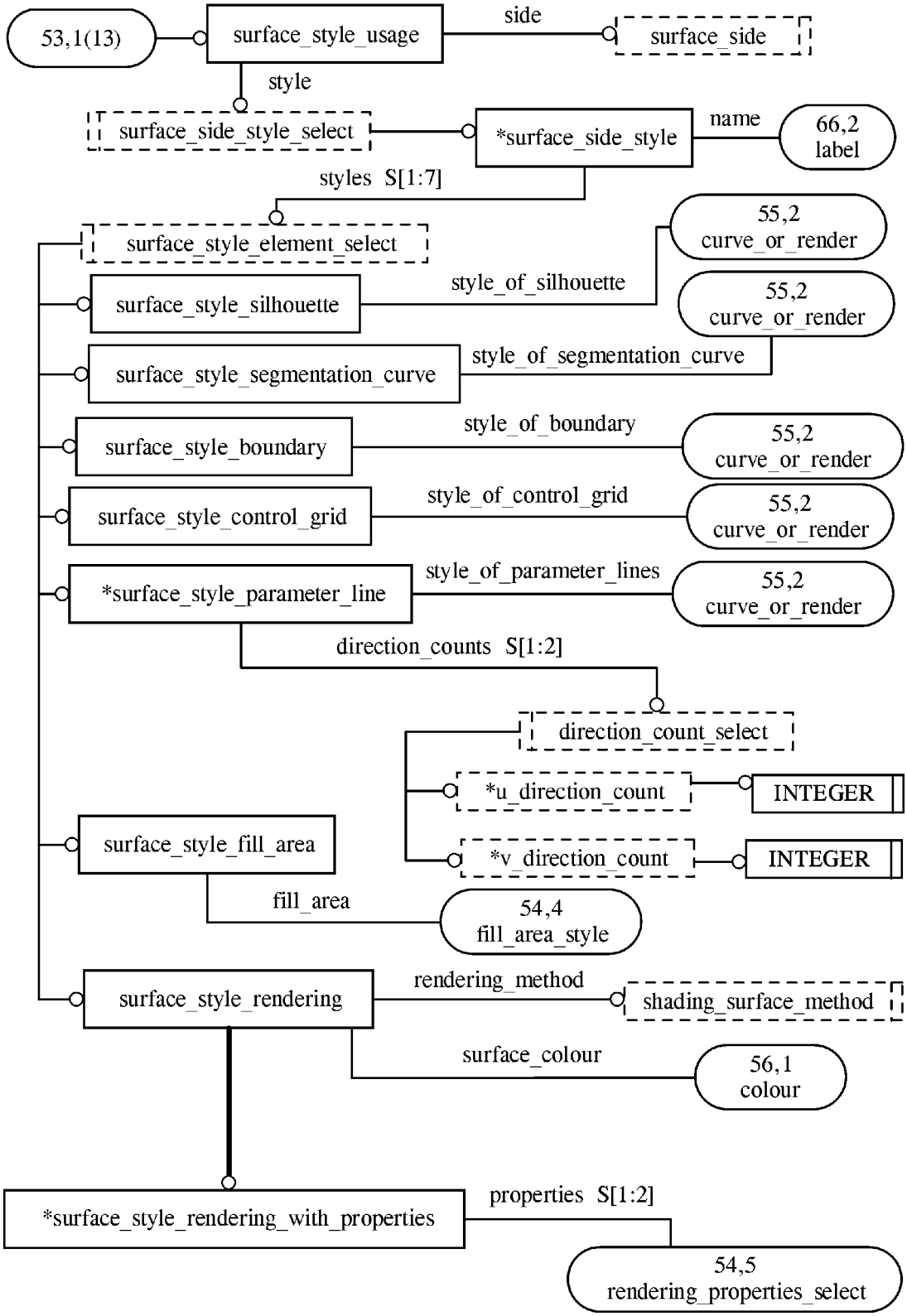

Figure H.53 — AIM EXPRESS-G diagram (53 of 66)

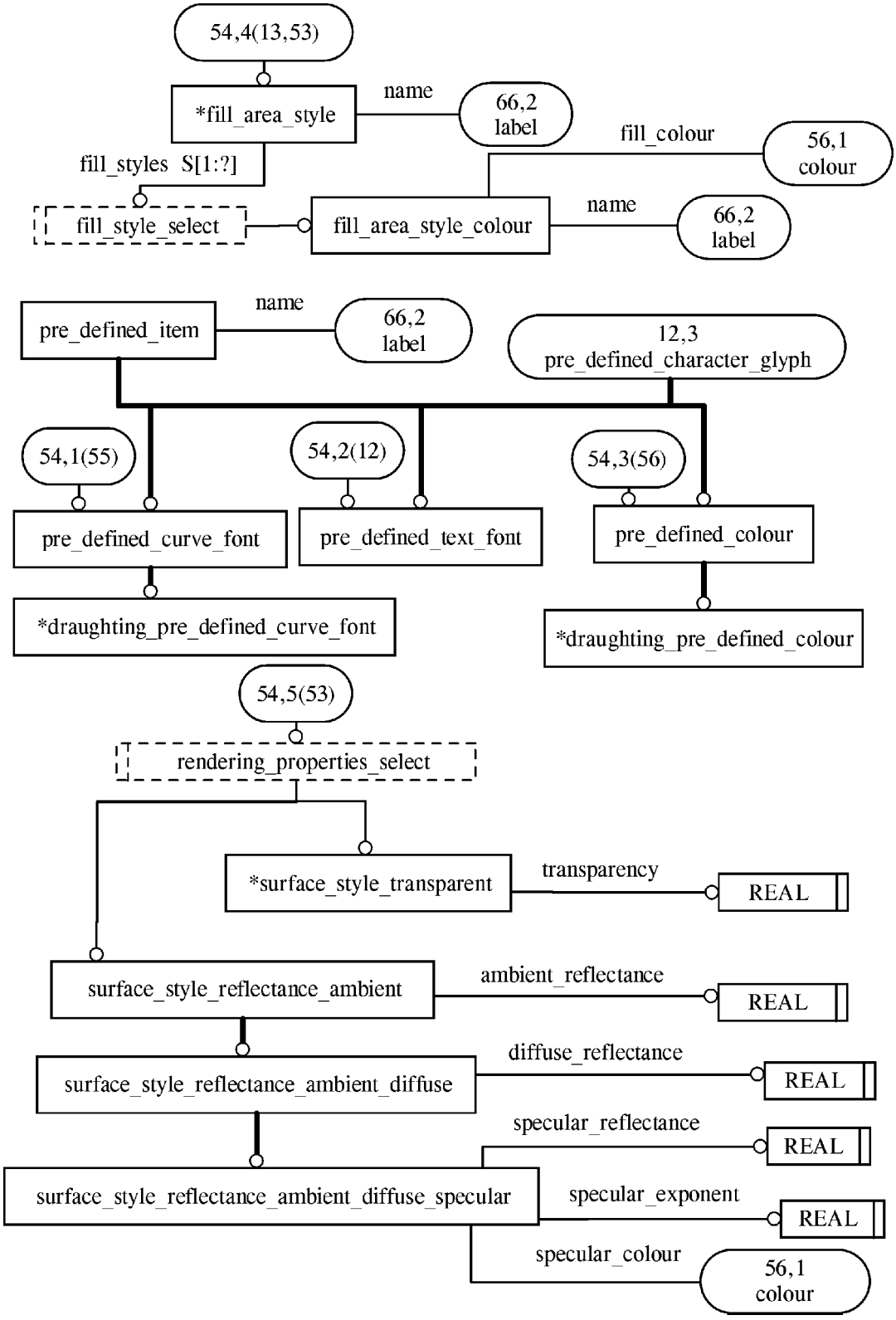

Figure H.54 — AIM EXPRESS-G diagram (54 of 66)

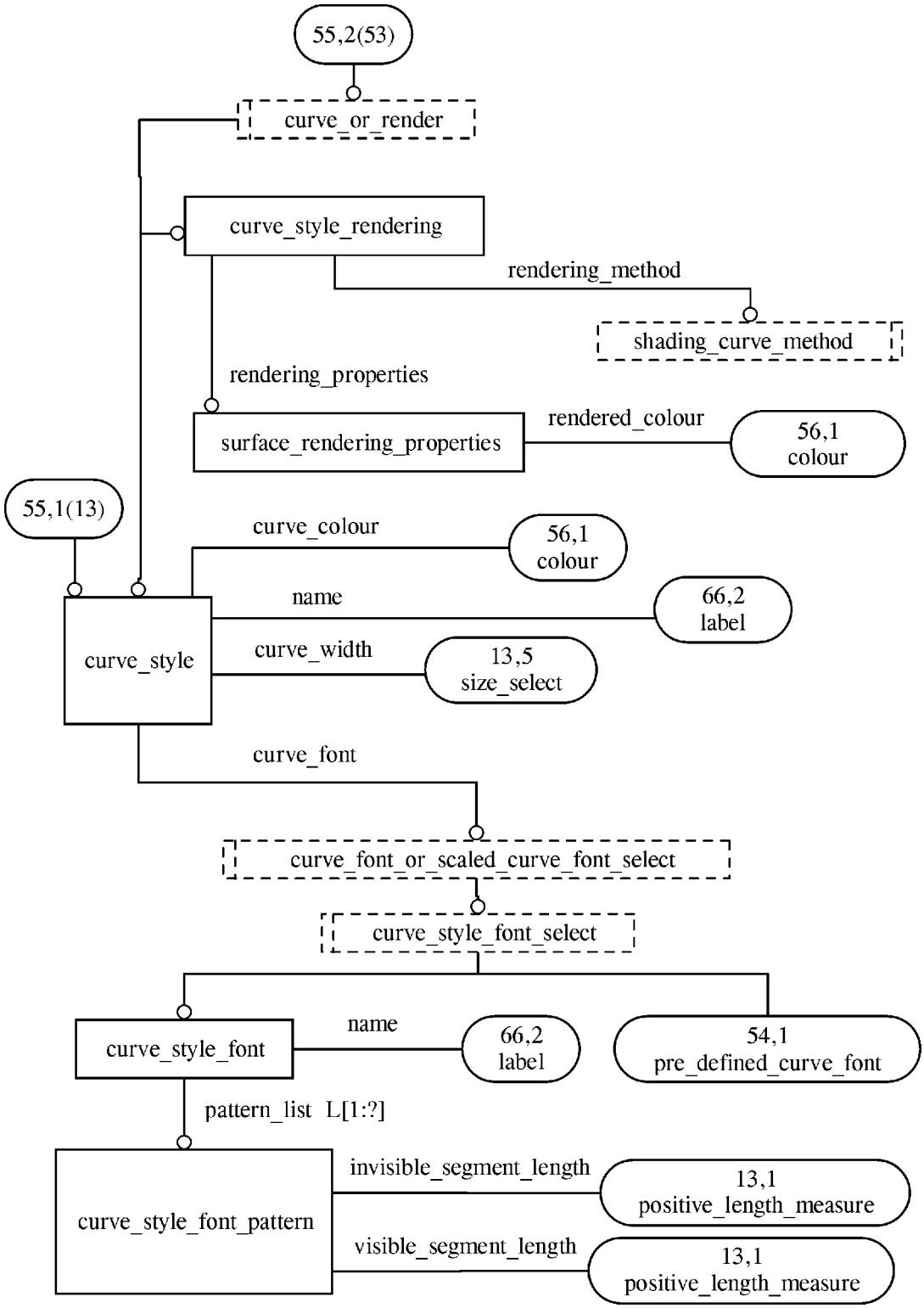

Figure H.55 — AIM EXPRESS-G diagram (55 of 66)

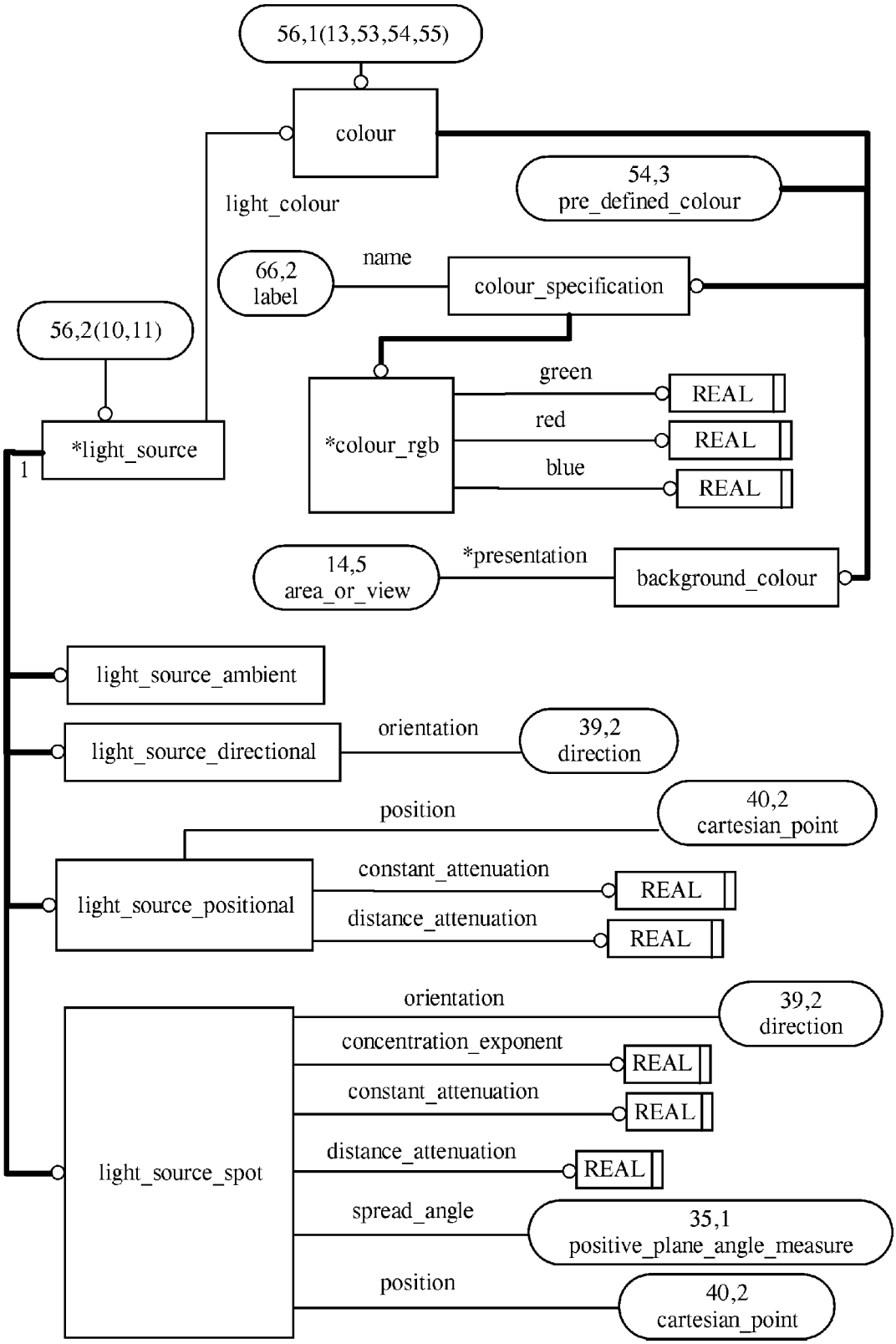

Figure H.56 — AIM EXPRESS-G diagram (56 of 66)

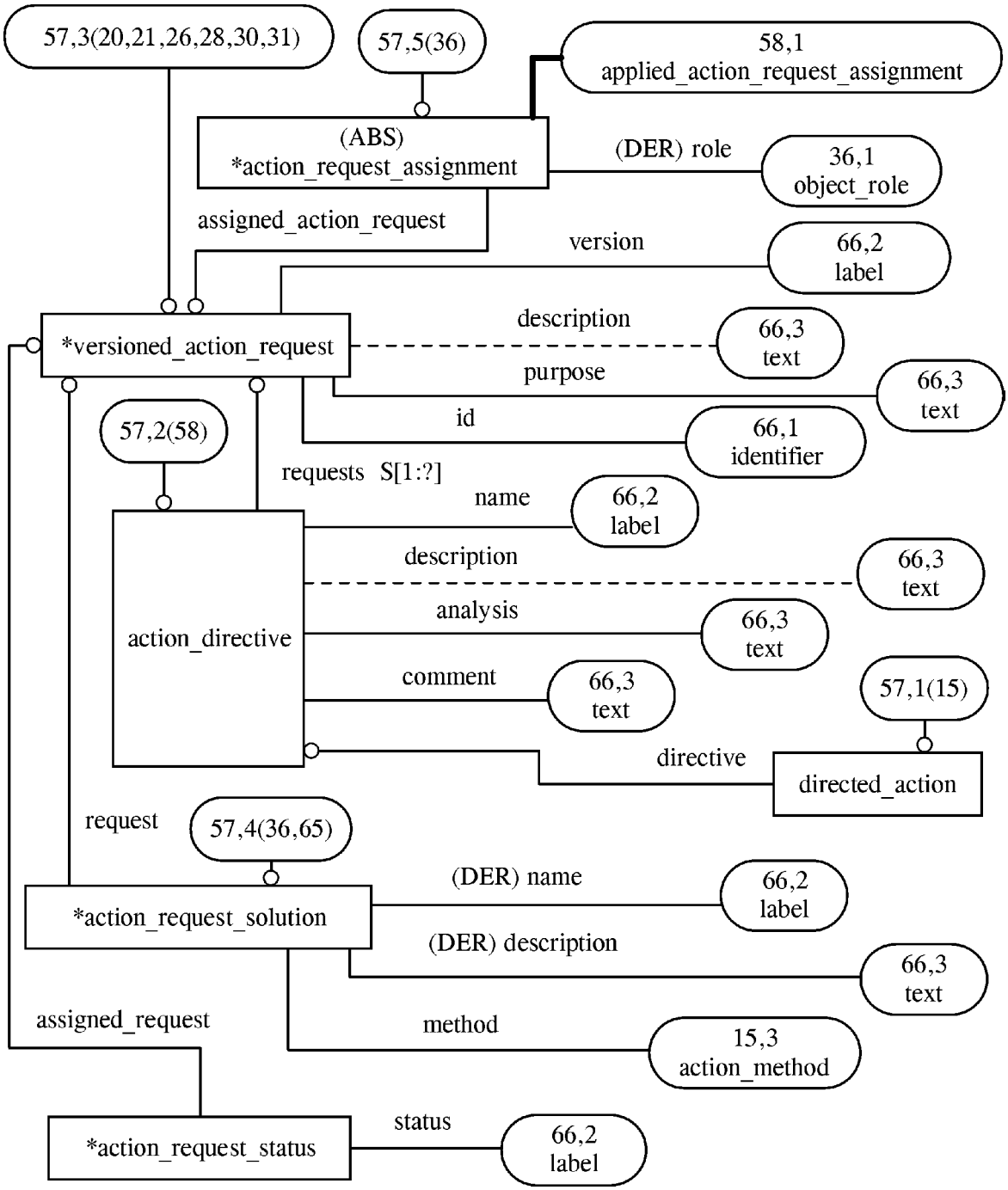

Figure H.57 — AIM EXPRESS-G diagram (57 of 66)

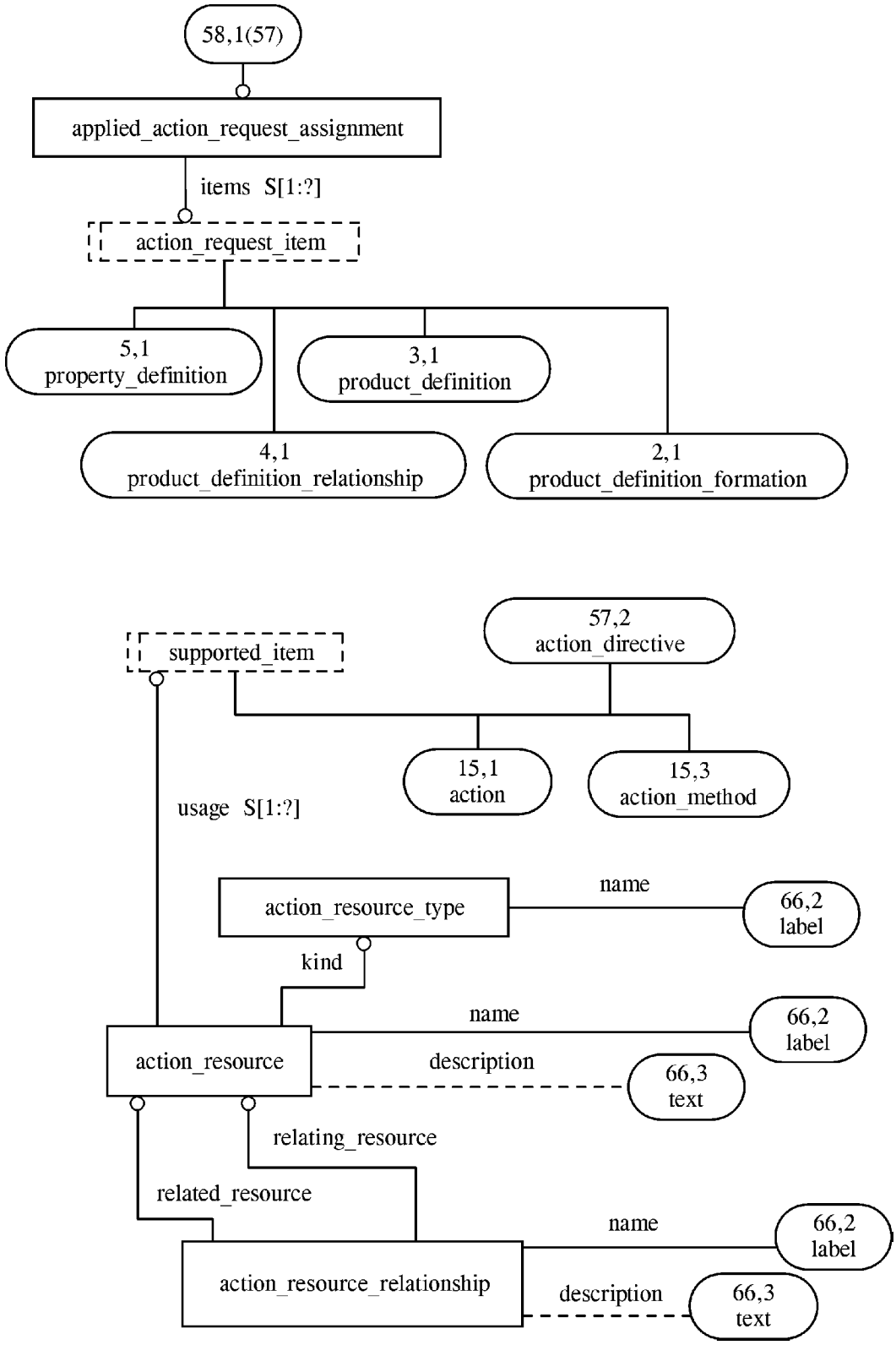

Figure H.58 — AIM EXPRESS-G diagram (58 of 66)

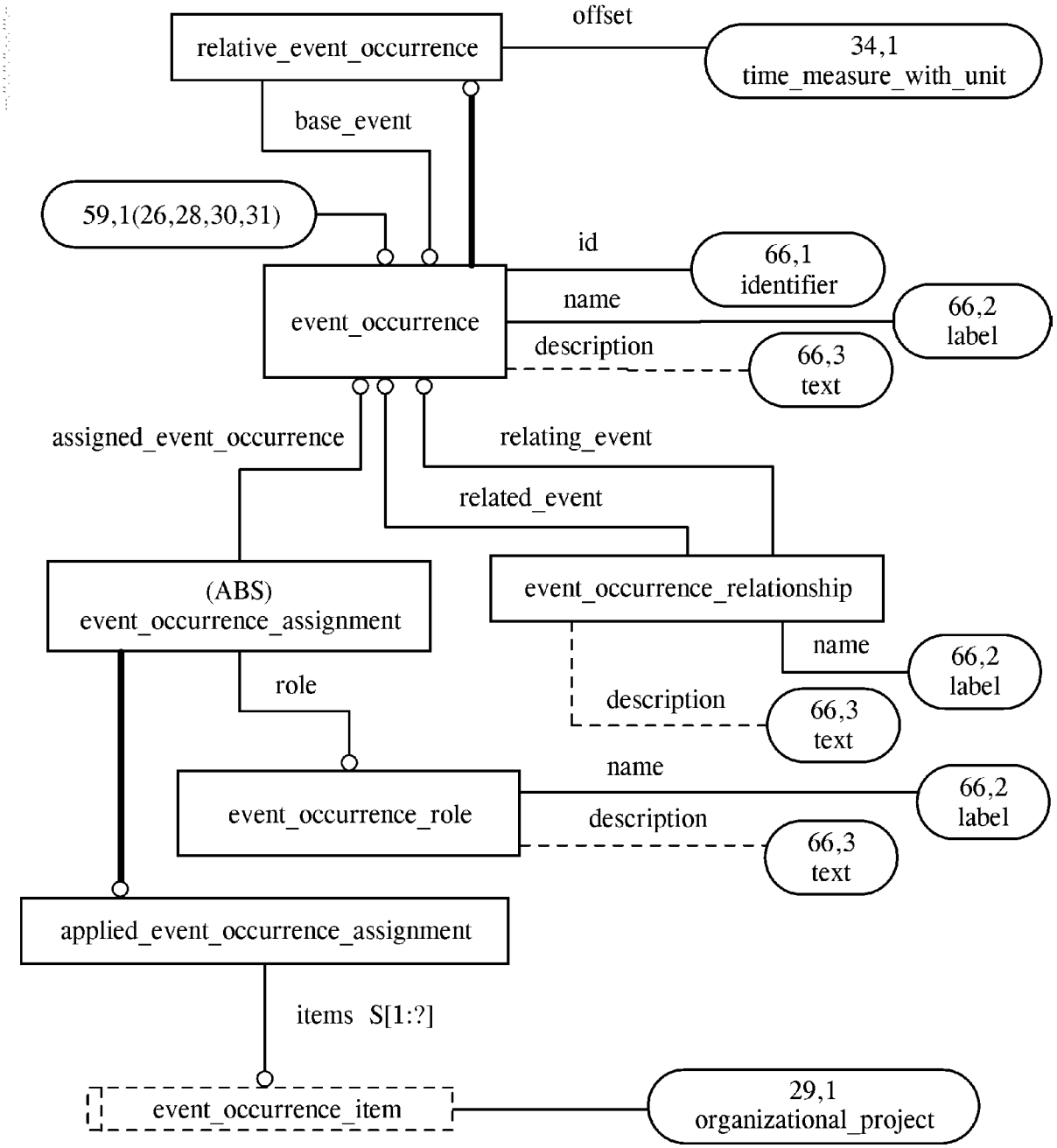

Figure H.59 — AIM EXPRESS-G diagram (59 of 66)

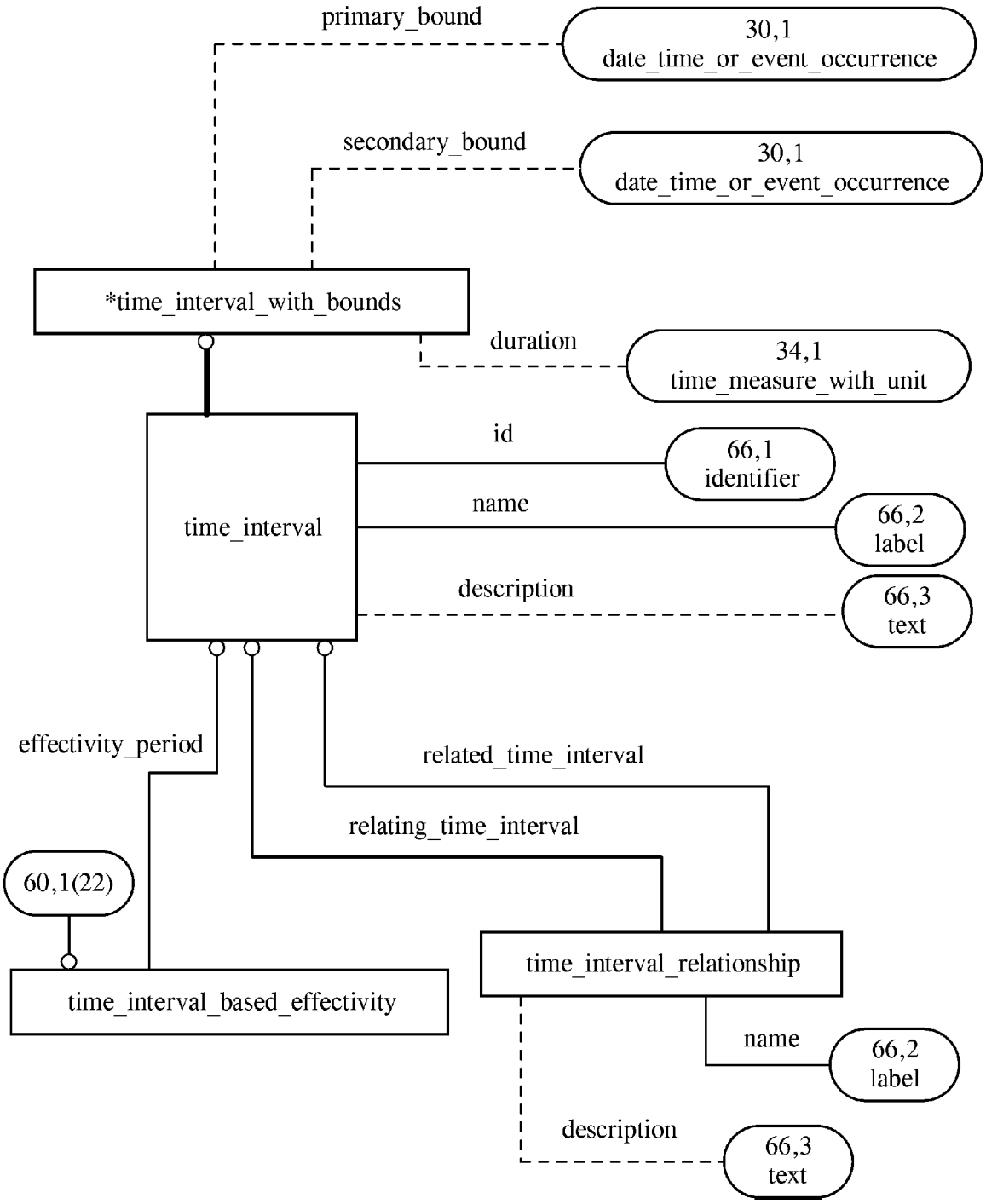

Figure H.60 — AIM EXPRESS-G diagram (60 of 66)

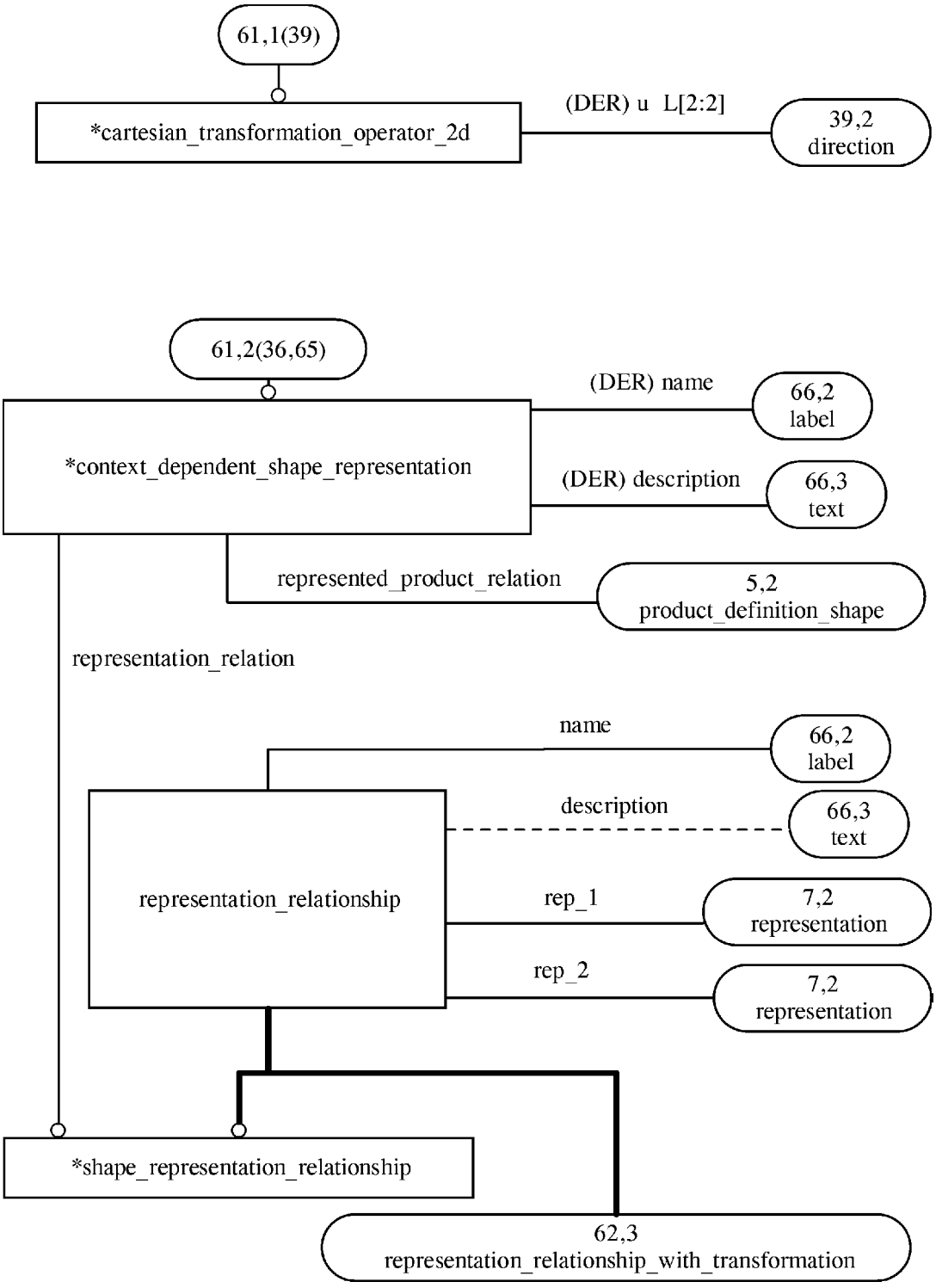

Figure H.61 — AIM EXPRESS-G diagram (61 of 66)

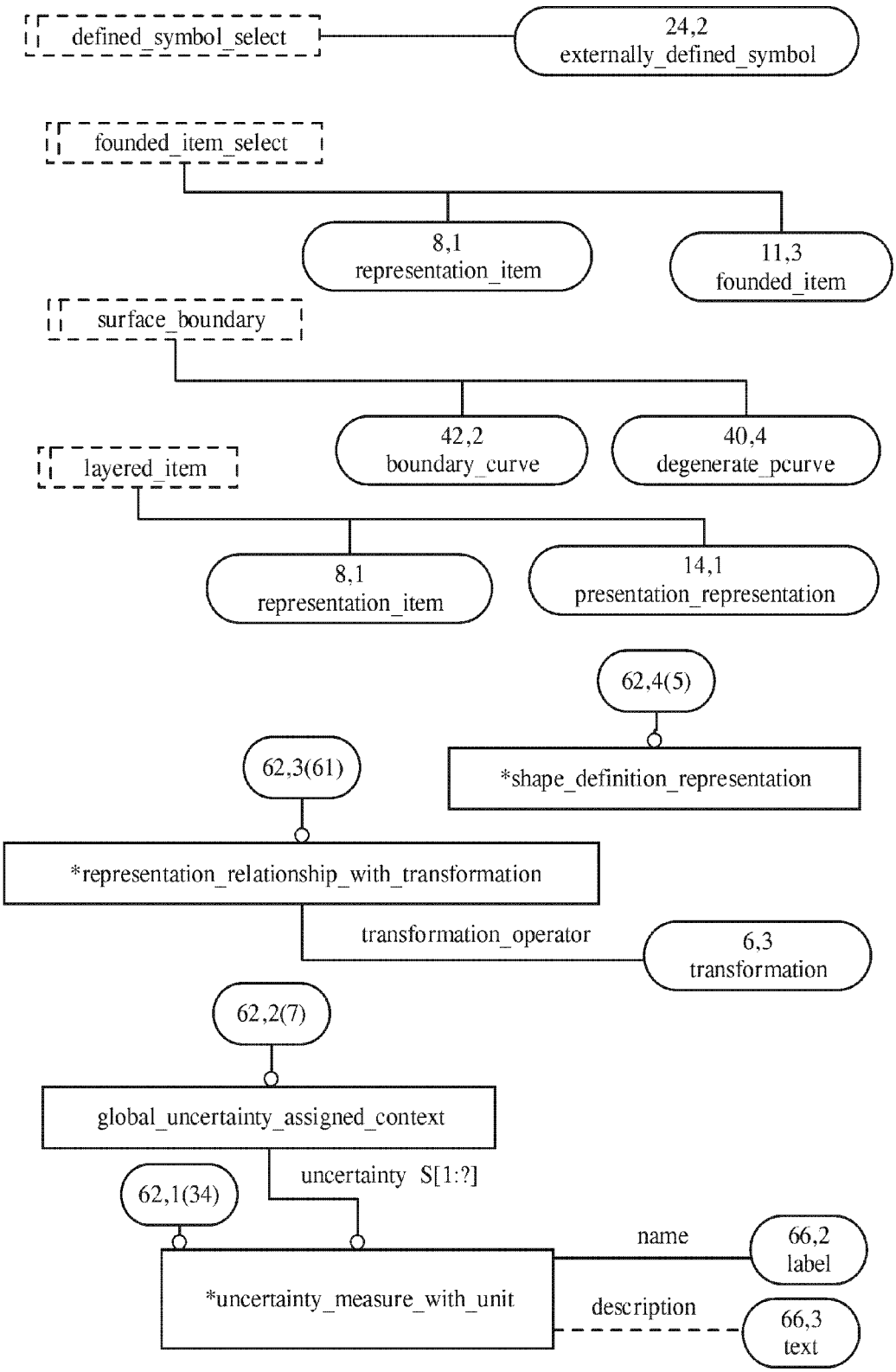

Figure H.62 — AIM EXPRESS-G diagram (62 of 66)

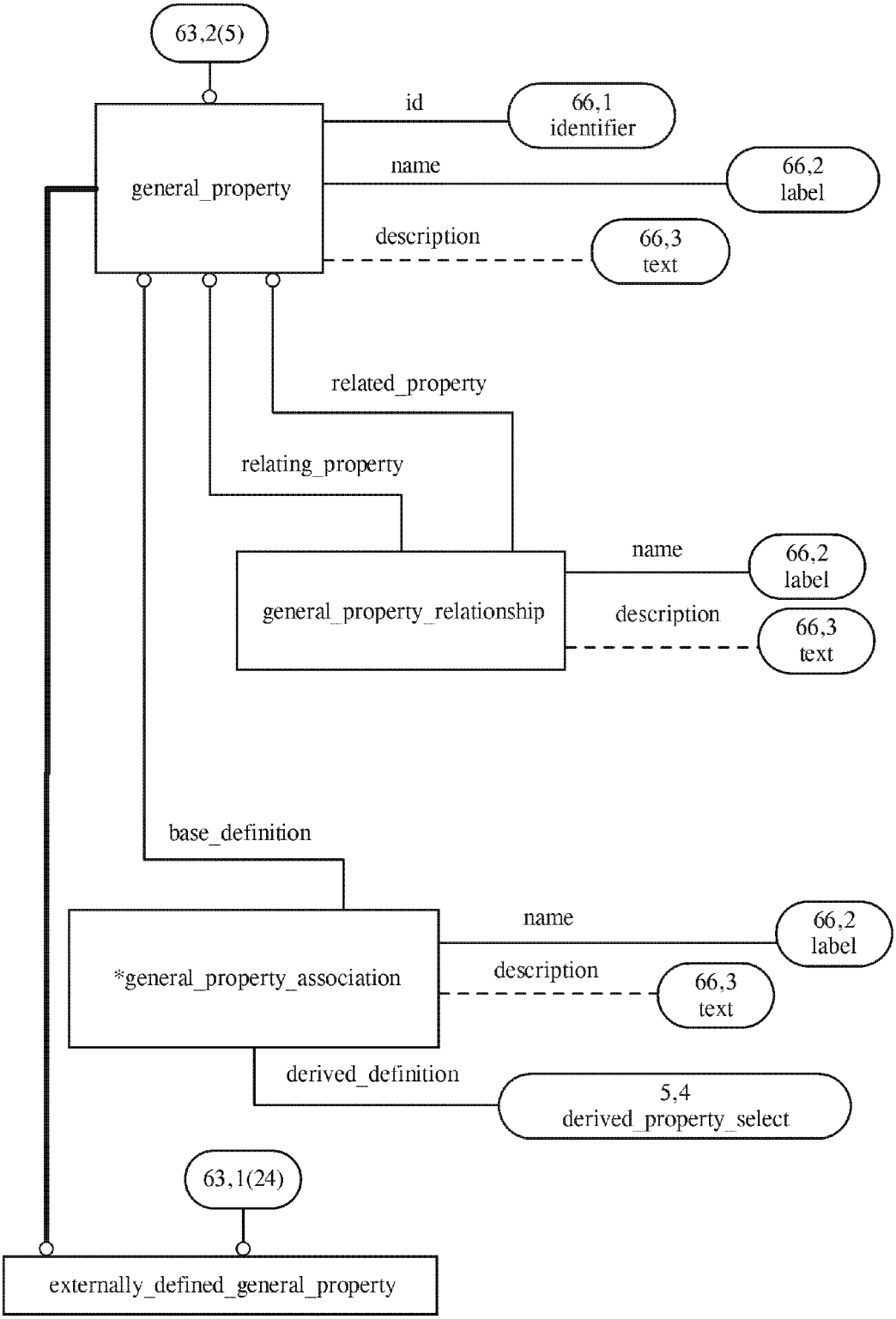

Figure H.63 — AIM EXPRESS-G diagram (63 of 66)

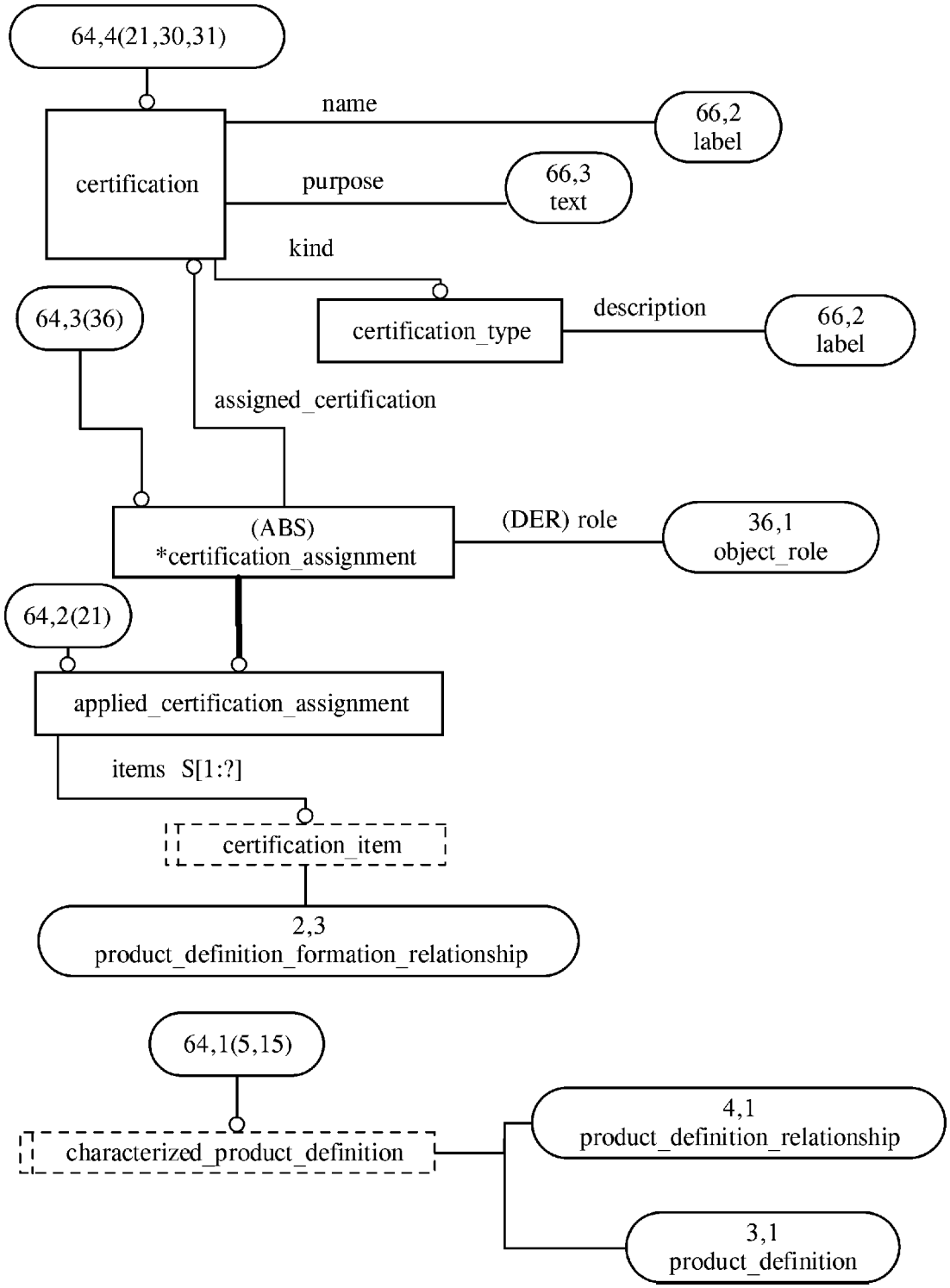

Figure H.64 — AIM EXPRESS-G diagram (64 of 66)

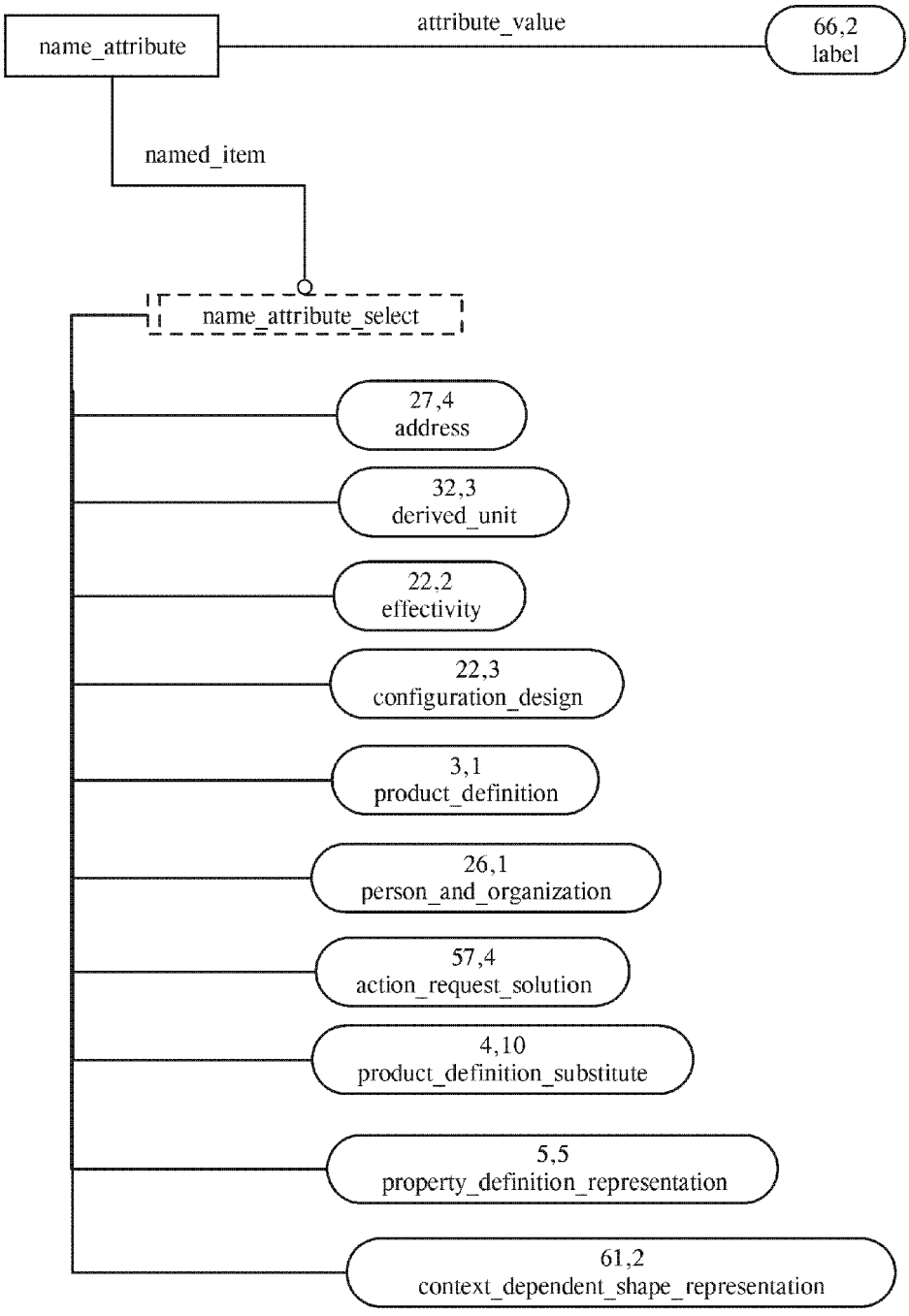

Figure H.65 — AIM EXPRESS-G diagram (65 of 66)

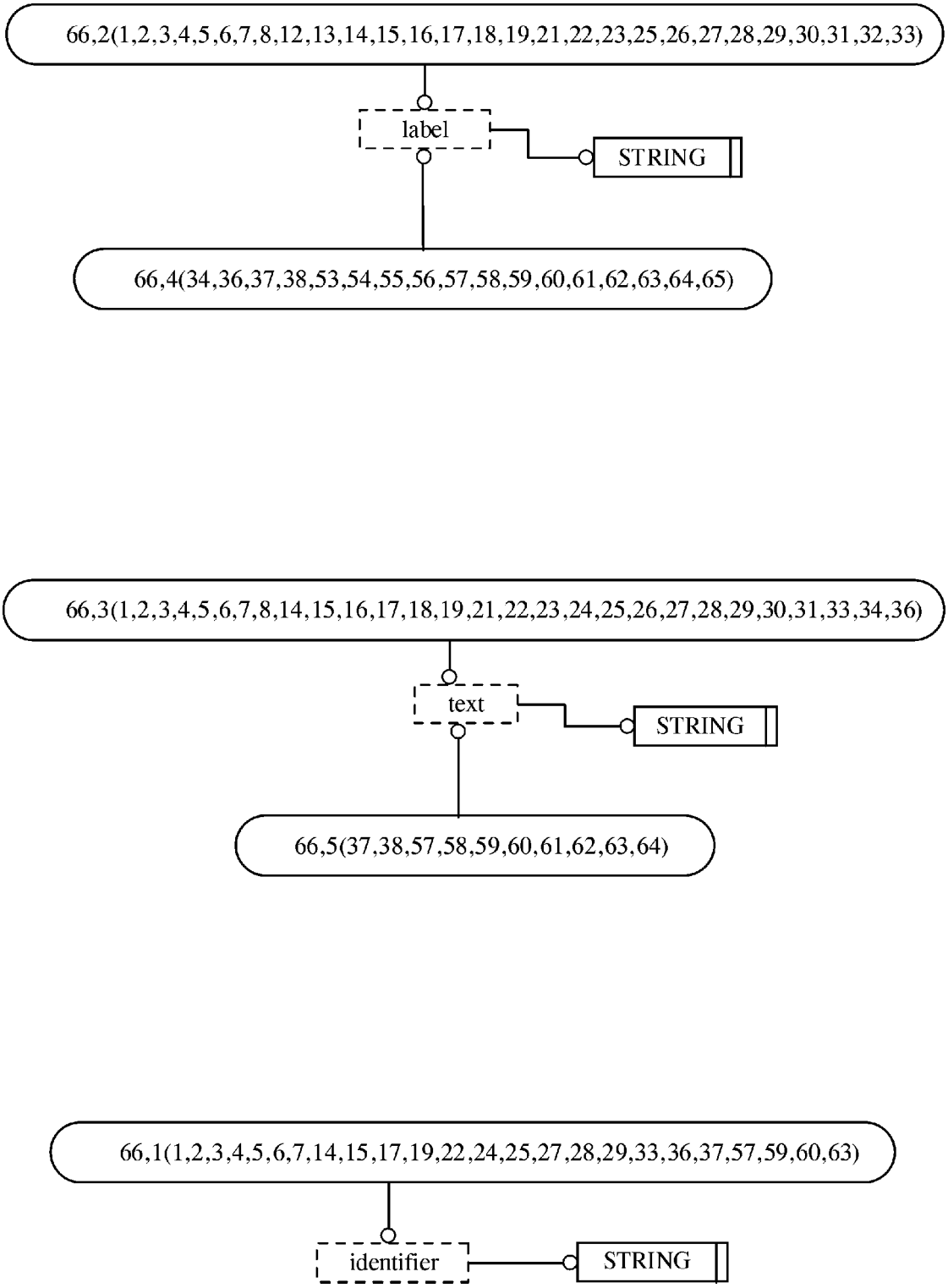

Figure H.66 — AIM EXPRESS-G diagram (66 of 66)

Annex J
(informative)

Computer-interpretable listings

This annex provides a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also provides a listing of the complete EXPRESS schema specified in this part of ISO 10303 without comments or other explanatory text. This annex is available in computer-interpretable form and can be found at the following URLs:

Short names: http://www.mel.nist.gov/div826/subject/apde/snr/
EXPRESS: http://www.mel.nist.gov/step/parts/part232/dis/

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex K
(informative)

Usage Scenarios

This annex has two major topics. The first topic, K.1, deals with describing information exchange scenarios. The second topic, K.2 through K.6 describes the different levels of capabilities that can be utilized within a unit of functionality. The units of functionality in this annex that have these levels of capabilities described for them are:

— data definition exchange;

— data list;

— index list;

— indentured data list;

— parts list.

K.1 Information exchange scenarios

Two information exchange scenarios have been created to highlight different aspects of how this part of ISO 10301 may be used. Product data exchange among PDM systems in different enterprises and the creation stage of the information that goes into formulating a Technical Data Package are the aspects of the following two scenarios, respectively. These two scenarios will show possible usages or business practices. Annex K defines capability levels of this part of ISO 10303 that can be applied to the following scenarios or any other scenario.

K.1.1 Design data information exchange scenario

An information exchange scenario was created for this part of ISO 10303 to highlight the exchange aspects among PDM systems. This scenario will show the functionality and the interoperability of this part of 10303 and other ISO 10303 Application Protocol parts, such as ISO 10303-209 [4], in a real world scenario. This scenario will simulate the business process of exchanging technical information among a prime contractor, design/analysis team, and end customer, as shown in Figure K.1. The scenario can be viewed as six stages. The stages are as follows:

— 'Prime Contractor' putting technical data package of requirements together and sending the package to 'Design/Analysis Team';

— 'Design Analysis Team' perform 'Preliminary sizing and design';

— 'Design Analysis Team' perform finite element analysis of design;

— 'Design Analysis Team' create 'Technical data package of sub-assembly' and ship to 'Prime Contractor';h

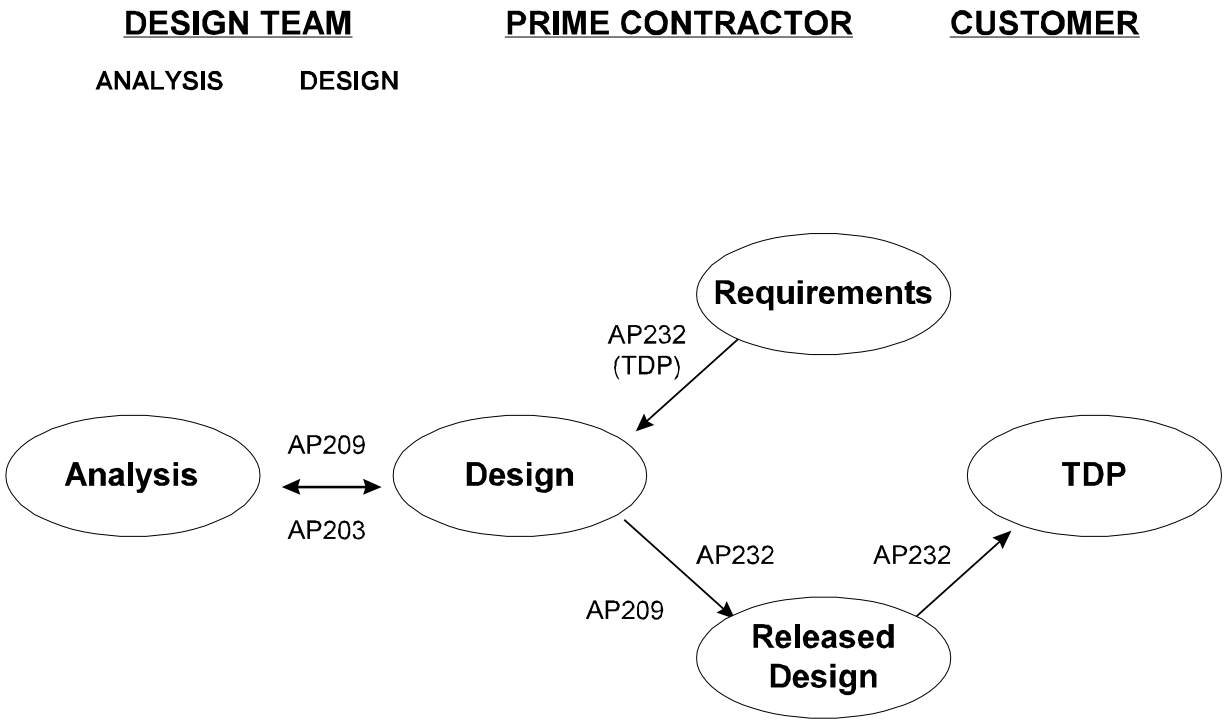

Figure K.1 — TDP information exchange activity

— 'Prime Contractor' incorporate technical data into overall technical data package and deliver to end customer;

— 'End Customer' incorporate technical data package into their data management system.

K.1.1.1 Prime Contractor - Packaging of technical requirements

The first business process stage deals with a 'Prime Contractor' packaging technical requirements for a sub-assembly and sending these requirements to a 'Design/Analysis Team'.

The 'Prime Contractor' will utilize the PDM's product and document structure to identify the out-sourced sub-assembly. Requirement data sets for sub-assembly are then identified and prepared for shipment to 'Design/Analysis Team'. Requirement data sets may include a procurement control drawing, sub-assembly shape envelope, additional technical requirements, and meta data that packages and relates these requirement data sets together. This meta data may be captured within this part of ISO 10303.

K.1.1.2 Design/Analysis Team - Preliminary sizing and design

The second business process stage deals with a subcontractor or teaming partner receiving the requirement data sets from the 'Prime Contractor', and performing preliminary sizing and design on the components of the sub-assembly. The subcontractor ot teaming partner is the 'Design/Analysis Team'.

The subcontractor would take the received data files and use the exchange meta data with the TDP to properly place them under configuration control. A product and document structure just for the sub-assembly would be created to reflect proposed product structure and data to be generated in the design/analysis process. This sub-assembly structure should identify the high level data that will be delivered to the Prime contractor. The sub-assembly shape envelope file, that is in an ISO 10303-21 file format utilizing ISO 10303-203, ISO 10303-209 [4], or this part of ISO 10303 format, will be translated to the Design/Analysis Team's native CAD System.

The sub-assembly shape envelope is then utilized to prepare shape parameter packets for individual sub-assembly components. The result of this activity provides shape information used as input to preliminary design activities such as initial structural sizing. Utilizing ISO 10303-209 [4], one or more of the resulting shape parameter packets for structural sizing will be sent to automated design tools. In these design tools, detail shapes and parameters of the component design will be formulated (for example, shape features, constituent ply shapes, and ply tables). This initial component design is then converted back into an ISO 10303-21 file for ISO 10303-209 [4] information.

K.1.1.3 Design/Analysis Team - Finite element analysis of design

The third business process stage deals with performing analytical structural validation of the component. A populated ISO 10303-209 [4]file will be utilized as a common storage location which facilities sharing and relating component data along different steps of the finite element analysis process.

Part shape via ISO 10303-209 [4] will be translated into a FEA pre-processor. A FEA model will be generated on the FEA pre-processor. This FEA will be translated and merged into the existing ISO 10303-209 [4] file. Material properties for each finite element will be generated utilizing the ISO 10303-209 [4] file as input. The material properties will be inserted into the FEA pre-processor and control information for the FEA model generated. The FEA controls are then translated and merged into the existing ISO 10303-209 [4] file. A FEA solver will be run against the FEA model in the FEA pre-processor. The results of the FEA solver will then be both translated and merged into the existing ISO 10303-209 [4]file and viewed by a FEA post processor.

At this point, an iteration on the design could occur if the FEA result warranted it.

K.1.1.4 Design/Analysis Team - Technical data package of sub-assembly

The fourth business process stage deals with performing formal documentation of the sub-assembly component and related the component's data within the sub-assembly's IDL and shipping list to Prime Contractor.

This consists of creating the parts list, data list, indentured data list, applications list, product data sets (containing product 3D shape information) and shipping list utilizing this part of ISO 10303. The drawing will be captured in this part of ISO 10303 or ISO 10303-202. In this scenario, any component and/or sub-assembly technical information may be created or captured in other file formats and incorporated into the technical data package for the sub-assembly delivery to the Prime Contractor.

A more in-depth look into the actual creation and formatting of the information in a released engineering technical data package is presented in subclause K.1.2, Application activity model information creation scenario.

K.1.1.5 Prime Contractor - Technical data package of end-item

The fifth business process stage deals with creating/updating the technical data package for the end-item that will be shipped to the 'End-Customer'.

The 'Prime Contractor' takes the technical data package from the 'Design/Analysis Team' and either updates the 'End-Item' product and document structure or maintains a separate sub-assembly product and document structure while placing the technical data in the appropriate places based on the 'Prime Contractor's' enterprise data configuration control system.

The 'Prime Contractor' then creates the packaging information needed to ship the updated 'End-Item' data to the 'Customer'. At this point, the 'Prime Contractor' could send a complete sub-assembly TDP, a partial TDP (reflecting just changes), or just the TDP meta data (describing what data is available for the sub-assembly). For this scenario, select a partial TDP consisting of information related to one component in the sub-assembly. Other sub-assemblies and components which make up the End-Item will be identified only with their product data being noted as "to be delivered at a later time". The creation of the shipping list data file, captured in the this part of ISO 10303, could be done through a Product Data Management (PDM) system or something equivalent.

K.1.1.6 End Customer - Storage and usage of technical data package

The fifth business process stage deals with the 'End Customer' receiving the technical data package from the 'Prime Contractor' and placing the received information in it's proper location in their system. Potentially a bid package could be constructed, utilizing this part of ISO 10303, that would identify and related the required technical data needed to reprocure a product.

K.1.2 TDP creation information exchange scenario

An information exchange scenario was created for this part of ISO 10303 to condense the TDP creation portion of the AAM activity model in this annex down to a single figure from the view point of the engineer that developed the TDP. This was accomplished so that this portion of the AAM could be explained without the level of detail that is required for the complete AAM. Node A2, Manage, Design, Build and Support a Product, is the portion of the AAM this scenario reflects. Figure K.2 delineates this single figure to present this portion of the abbreviated AAM.

Typical TDP documentation starts with the development of the design in whatever form/format is required. This is classically a two-dimensional geometric format, but is more frequently being developed in a three-dimensional geometric environment. The TDP documentation process follows with determining the drawing requirements for the respective customer and translating that into the required things such as drawing border, format, symbology, and terminology of the customer (and/or seeking a deviation from the respective customer requirements) for drawing based TDP practices. The TDP documentation process for product data sets define product data set presentation and format requirements for the respective customer and determining the ability to satisfy these requirements with current data exchange standards and specifications.

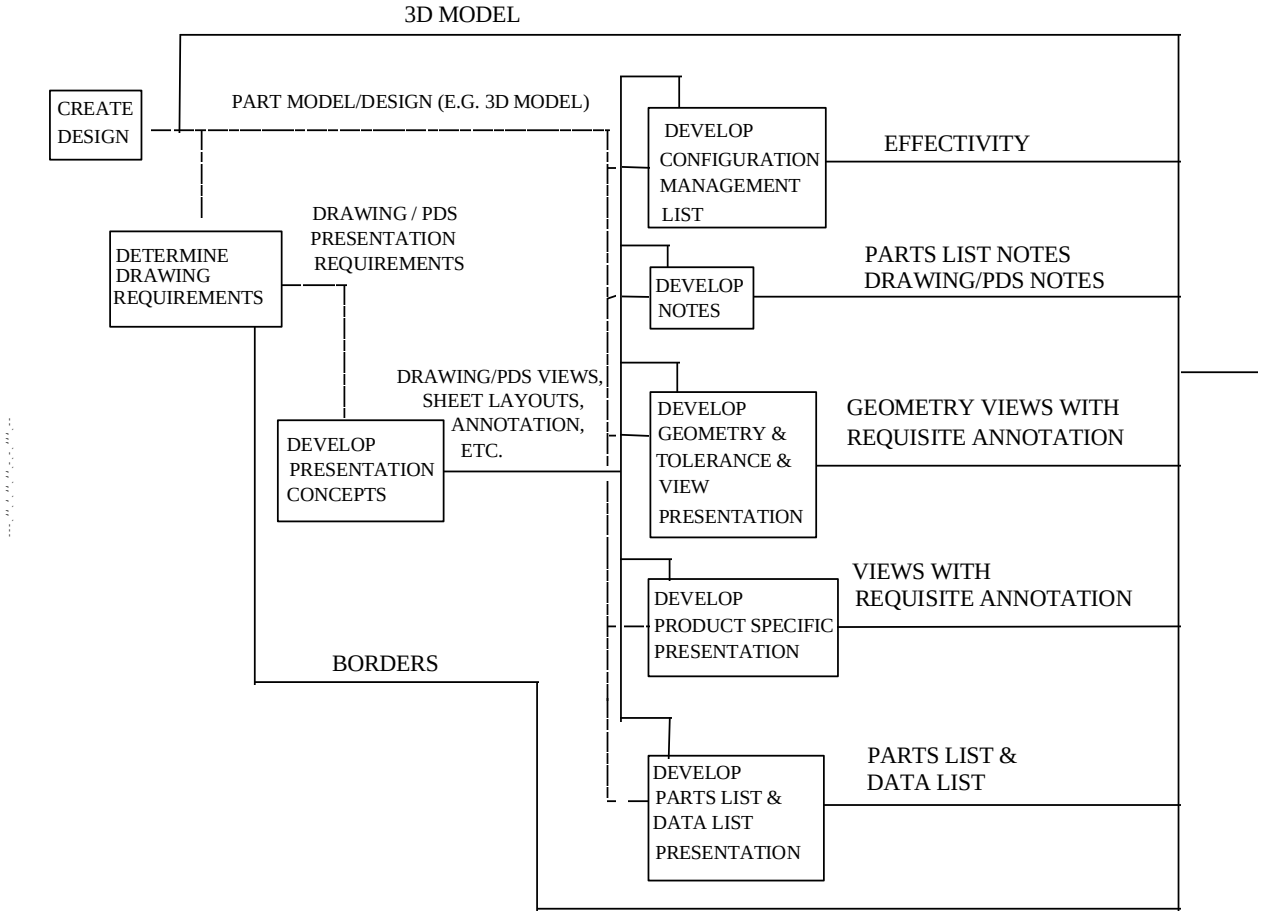

Figure K.2 — Information creation scenario for TDP development

The customer requirements will dictate what the presentation concepts are for the geometric and textual information. The geometric information that was utilized in the creation of the design is then converted to the appropriate form/format that is required to appropriately document the design. This is done in several stages that are delineated in the five boxes on the right hand side of Figure K.2. These activities would be done manually within a computer-aided design application and the output would be translated into the appropriate design format for exchange to the downstream applications.

Information that would not be captured in this scenario is a direct relationship between a piece of geometric information and the annotation (including dimensions and tolerances) required to document the geometric information. A relationship is established between the views of the geometric information, but these views would not have the geometry in the view related to one another. The view data is directly related to the three-dimensional model, but this is not on a geometric element level. The relationship is at a higher level than the geometric element level. The three-dimensional data is not required, but is preferred. The master geometric representation is the two-dimensional view of the data with the three-dimensional data as reference material for the application.

Associated List information (for example, Parts List, Data Lists) is maintained at a level of capability so that an automated application could process the information in multiple ways. These associated lists would be maintained as separate documents.

All this product data is then associated together utilizing some form of Product Data Management (PDM) system.

The following terms are used in the TDP creation information exchange scenario in Figure K.2.

The definitions given in this annex do not supersede the definitions given in the main body of the text.

K.1.2.1
Create Design
Develop design concepts of shapes, fits, and interrelationships to other parts and structural requirements. This includes possible variant design, special design applications, and investigation of downstream component requirements.

K.1.2.2
Determine Drawing Requirements
Determine the appropriate drawing documentation requirements or Product Data Set preparation requirements for the component. This includes review of company and customer drawing /PDS reference manuals and standards.

K.1.2.3
Develop Configuration Management List
Verify the configuration of the component under design and document appropriate effectivity of the component. This would include update of the configuration information for a modification of the component.

K.1.2.4
Develop Geometry & Tolerance & View Presentation
Document the shape of the component in a manner whereby a competent manufacturer could produce the component from the design documentation. This includes manufacturing tolerances.

K.1.2.5
Develop Notes
Document the required textual information. This would include process, drawing/PDS interpretation, finish, material and other requirements.

K.1.2.6
Develop Part List & Data List Presentation
Document the required list information in the appropriate formats that comply with company/customer standards for presenting face of the drawing data. This includes the reference between the shape information and the parts list and/or data list. This reference is not explicitly represented in a digital manner.

K.1.2.7
Develop Presentation Concept
Develop the appropriate presentation concepts for the component to adequately represent the design on the drawing/PDS. This includes the view of the component on the drawing/PDS including cross-sectional views, as well as required textual information for things such as notes and part lists.

K.1.2.8
Develop Product Specific Presentation
Document the appropriate shape information and requisite annotation so that downstream applications can utilize the shape information without recourse to the original design activity.

The following give the definitions of the data that is flowing through the activities.

K.1.2.9
3-D Model
This is the three-dimensional geometric representation of the component.

K.1.2.10
Borders:
This is the standard drawing border that is selected to contain the design documentation.

K.1.2.11
Drawing/PDS Presentation Requirements:
These are the presentation requirements for the part model /design in the drawing/PDS.

K.1.2.12
Effectivity
This is the properly documented effectivity of the component on a drawing/PDS and within the appropriate list.

K.1.2.13
Geometry Views with Requisite Annotation
These are the view plane representations of the geometry and the annotation required to delineate the design requirements and allowances for downstream applications (such as, tolerances and view dependent information.).

K.1.2.14
Part Model/Design
This is the shape representation and associated textual information of the component in whatever form is appropriate to utilize in documenting the design in a drawing/PDS.

K.1.2.15
Parts List & Data List
These are the lists that are required for the design documentation. These lists delineate the appropriate documentation for the component so that downstream applications can build the component.

K.1.2.16
Parts List Notes, Drawing/PDS Notes
These are any required notes on the Parts List (for example, finish, heat treat.) or Drawing /PDS (such as, interpretation standards and revision.) that are required for downstream applications to build and/or support the component.

K.1.2.17
Product Specific Views with Requisite Annotation
These are views of the geometric representation and the requisite annotation which document specific product characteristics (for example, cross sectional views, schematics, composite ply layups).

K.1.2.18
Views, Sheet Layouts, Annotation
Views are the selected shape views of the component that will be documented in a drawing/PDS. Sheet Layouts are the sheet size and sheet constraints that are applied to document the design with the given drawing requirements. Annotation is the drawing/PDS requirement for presenting the annotation in the design documentation.

K.2 Definitions and levels of capability for the parts list

This appendix section defines the Parts List (PL) and its associated information. The levels of capability for the PL focus on the ability to capture part or item information including configuration and accountability information for the purposes of exchanging the PL information between two enterprises or between two systems. These levels of capability do not address graphical presentation of the Parts List.

K.2.1 Parts list definition

The Parts List is a tabulation of parts and bulk materials (except those materials that support a process) when used in the item. Reference documents may also be tabulated on parts lists. Items listed on subordinate assembly parts lists or specified in a referenced document are not repeated in the using assembly parts list unless it is necessary to limit options permitted by the subordinate document. In-house documents, for in-house usage only may be referenced parenthetically.

K.2.2 Parts list levels of capability

The functional utility of the PL is dependent on the amount of TDP element information recorded. The following levels of capability define an increasing level of utility attributed to the number of application objects used to comprise it.

K.2.2.1 Minimal level

This level provides a minimum level of item or part identification capability for commercial exchanges. The PL constructs identified allow for simple item relationships and characteristics to be exchanged. Only mandatory application objects are identified. All other application objects in the following levels of capability are considered optional in a commercial exchange. This level of capability allows for the capture and exchange of the following information:

— identifies the PL document number and version;

— identifies the PL version and number of sheets;

— identifies the find number of a part on the drawing (when applicable);

— identifies notes or remarks pertinent to the item or part in the PL tabulations;

— identifies parts (or identifying numbers) by number, nomenclature (such as, name), and quantity required for the item, such as, assembly.

K.2.2.2 Controlled level

This level provides a higher level than the minimum level of part or item identification capability.

Government requirements are usually more stringent than the commercial requirements in the definition of the developing design authority information and in configuration procedures for the PL.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities identified in minimum level above;

— identifies the contract number under which the PL was prepared;

— identifies the title of the PL;

— identifies the preparing activity of the PL and the preparing activity of the tabulated entries in the PL;

— identifies the weight of the item or part;

— identifies the release authentication (when required).

K.2.2.3 Limited history and material identification level

This level of capability contains two different aspects. The first aspect is the capability to identify a particular configuration of a part that is applicable to a specific effectivity of a product (such as, limited history). The other capability is to identify material information and material stock characteristics for a part or item. This level of capability allows for the capture and exchange of the following information:

— all of the capabilities in the controlled level of capability;

— additional information for material identification that includes stock material, stock size, and document identifying material characteristics;

— limited history for identification of the part or item into the end item configuration. This includes quantity of the part or item in the end item.

K.2.2.4 Supplemental identification level

The supplemental identification level of capability provides capabilities to those previously cited in identifying and defining parts or items, limited history, and material. This level of capability also provides for:

— all of the capabilities in the limited history and material identification level;

— additional information for material identification that includes stock material, stock size, and document identifying material characteristics;

— limited history for identification of the part or item into the end item configuration. This includes quantity of the part or item in the end item;

— notation for each item or part, including special conditions identified for the item or part;

— security, distribution, and data rights information on a per item or part basis.

K.2.3 Parts list requirements traceability

Table K.2.1 and Table K.2.2 identify the application objects for the Parts List (PL) and to which level of capability they apply. The columns in the matrix are correlated to the levels of capability as follows:

— Column 1 is the application object, enumeration, or select;

— Column 2 = Level 1 = Minimum level (commercial);

— Column 3 = Level 2 = Controlled level (government);

— Column 4 = Level 3 = Limited history and material identification level;

— Column 5 = Level 4 = Supplemental identification level;

— Column 6 is notes of interest to the other columns.

Table K.2.1 defines the levels of capability for PL EXPRESS constructs for selects and enumerations. Table K.2.2 defines the levels of capability for PL applications objects.

Table K.2.1 — Levels of capability for PL selects and enumerations

Parts list Selects and enumerations	Levels of capability				Notes
	1	2	3	4	
Enumerations					
change_type				X	
component_list_type_enumeration				X	
retrofit_state				X	
quantity_accuracy_enumeration	X	X	X	X	
stock_size_classification			X	X	For material level
stock_size_cross_section			X	X	For material level

Table K.2.1 — Levels of capability for PL selects and enumerations (concluded)

Parts list Selects and enumerations	Levels of capability				Notes
	1	2	3	4	
weight_derivation				X	
Selects					
alternate_identification_element_select				X	
alternate_identification_item_select				X	
component_list_type_select	X	X	X	X	
date_or_event				X	
drawing_or_product_data_set				X	
geometry_select				X	
item_identifying_number_select	X	X	X	X	
item_list_applies_to_select	X	X	X	X	
quantity_accuracy_select	X	X	X	X	
quantity_type_select	X	X	X	X	
revision_authorization_select	X	X	X	X	
shape_select				X	
usage_condition				X	
usage_context					
weight_derivation		X	X	X	

Table K.2.2 — Levels of capability for PL application objects and attributes

Parts list objects	Levels of capability				Notes
	1	2	3	4	
a_number	X	X	X	X	
a_real	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
alternate_element_identification	X	X	X	X	
change_status			X	X	
design_activity	X	X	X	X	
element_classifications		X	X	X	
identifying_number	X	X	X	X	
outstanding_changes			X	X	
title	X	X	X	X	
alternate_identification_of_item	X	X	X	X	
change_status			X	X	
classifications			X	X	
design_activity	X	X	X	X	
identifying_number	X	X	X	X	
nomenclaature_or_name	X	X	X	X	
source_information				X	
alternate_item				X	
preference_order				X	
interchange_item				X	
type_of_alternate				X	
usage_conditions				X	
an_integer	X	X	X	X	
approval			X	X	For limited history
approval_date			X	X	For limited history
person_organization			X	X	For limited history

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
purpose_of			X	X	For limited history
assembly_relationship			X	X	For material identification
associated_list	X	X	X	X	
presentation				X	This is for presentation of the information only.
related_to				X	
certification				X	
approvals				X	
creation_date				X	
description				X	
name				X	
change_identification	X	X	X	X	
action_status				X	
assigned_to				X	
change_code				X	
change_date				X	
change_description				X	
change_level				X	
issue_date				X	
issue_level				X	
revision_authorization_identifications	X	X	X	X	
revision_date	X	X	X	X	
revision_description		X	X	X	
revision_level	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
company	X	X	X	X	
company_code		X	X	X	
code		X	X	X	
code_administrator			X	X	
type_of_code		X	X	X	
configuration	X	X	X	X	
data_usage_rights				X	
distribution_authorizations				X	
end_item_system_designation			X	X	Level 3 for limited history
preparing_contracts		X	X	X	
release_authorizations	X	X	X	X	
security_identificatoins				X	
contract		X	X	X	
affected_organizations				X	
approvals				X	
contract_data_requirements_list				X	
contract_number		X	X	X	
creation_date				X	
data_item_description_identification				X	
date	X	X	X	X	
day	X	X	X	X	
month	X	X	X	X	
specific_time	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
week	X	X	X	X	
year	X	X	X	X	
date_effectivity			X	X	For limited history
end_date			X	X	For limited history
start_date			X	X	For limited history
design_authority	X	X	X	X	
design_activity_code				X	
design_activity_identification	X	X	X	X	
person_responsible				X	
role				X	
distribution_notice				X	
distribution_authority				X	
distribution_code				X	
distribution_statement				X	
document_usage_parameter			X	X	
subject			X	X	
value_of			X	X	
drawing				X	
heading				X	
pages				X	
drawing_suffix_number_combination	X	X	X	X	
drawing_number	X	X	X	X	
suffix_identifier	X	X	X	X	if required

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
effectivity			X	X	for limited history
description			X	X	for limited history
effectivity_approvals				X	
name				X	
product				X	
work_activity				X	
element_identification	X	X	X	X	
alternate_identifications	X	X	X	X	
change_status	X	X	X	X	if applicable
design_activity	X	X	X	X	
element_certification				X	
element_classifications		X	X	X	
identifying_number	X	X	X	X	
outstanding_changes				X	
title		X	X	X	
element_type		X	X	X	
code_administrator				X	
element_code		X	X	X	
type_of_coding_scheme		X	X	X	
event				X	
actual_date				X	
assignment				X	
description				X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
id				X	
name				X	
planned_date				X	
responsible_person_organization				X	
external_library_reference				X	
description				X	
external_id				X	
library_type				X	
file				X	for digital mockup
file_format				X	
file_relationship				X	
foreign_defined_item			X	X	For material capability
general_item_definition_relationship				X	
header	X	X	X	X	
document_abstract		X	X	X	
document_keywords		X	X	X	
header_configuration	X	X	X	X	
language		X	X	X	
sheet_count	X	X	X	X	
size_of_sheet	X	X	X	X	
header_configuration_with_element_-identification	X	X	X	X	
data_configuration	X	X	X	X	
identification	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
identifier	X	X	X	X	
independent_property				X	
allowed_units				X	
description				X	
id				X	
property_source				X	
independent_property_relationship				X	
description				X	
related_property				X	
relating_property				X	
relation_type				X	
independent_property_usage				X	
assigned_property				X	
assigned_to				X	
item	X	X	X	X	
alternates				X	
conditions_defined_through_-constrained_document			X	X	For material capability
conditions_defined_through_simple_-reference			X	X	For material capability
identification	X	X	X	X	
item_configuration				X	
item_contexts	X	X	X	X	
item_weight	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
notes				X	
shape				X	
shape_aspects				X	
special_conditions				X	
item_identification	X	X	X	X	
alternate_identifications	X	X	X	X	
change_status	X	X	X	X	
classifications	X	X	X	X	
design_activity	X	X	X	X	
identifying_number	X	X	X	X	
item_certification				X	
nomenclature_or_name	X	X	X	X	
source_information					Not used
item_list	X	X	X	X	
component_of	X	X	X	X	
list_for	X	X	X	X	
type_of_component_list	X	X	X	X	
item_list_drawing	X	X	X	X	
assembly_item_list_applies_to	X	X	X	X	
drawing_number	X	X	X	X	
item_type	X	X	X	X	
code_administrator				X	
item_code	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
type_of_coding_scheme				X	
item_usage	X	X	X	X	
being_defined_for	X	X	X	X	
effective_on			X	X	For limited history capability
list_item	X	X	X	X	
quantity_used_in_next_higher_assembly	X	X	X	X	
retofit				X	
special_conditions		X	X	X	
label	X	X	X	X	
list_item_usage	X	X	X	X	
approvals				X	
drawing_zones				X	
entry_item_change_level	X	X	X	X	
entry_notes			X	X	
find_number				X	
item_type				X	
reference_designators				X	
security_information				X	
locally_defined_item		X	X	X	For drawing and material capability
location_instance				X	
geometry_definition				X	
id				X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
scale				X	
transformation_definition				X	
lot_effectivity			X	X	For limited history capability
lot_identification			X	X	For limited history capability
lot_size			X	X	For limited history capability
lot_unit_of_measure			X	X	For limited history capability
made_from_stock_material		X	X	X	For drawing and material capability
quantity_of_item_made_from _stock			X	X	For material capability
size					
specified_by		X	X	X	For drawing and material capability
stock			X	X	For material capability
make_from_relationship			X	X	For material capability
item_that_is_acting_as_stock			X	X	For material capability
quantity_of_item_made_from_other_-item			X	X	For material capability
measure	X	X	X	X	
notation	X	X	X	X	
note	X	X	X	X	
note_parameter	X	X	X	X	
note_title	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
reference_code	X	X	X	X	
referenced_document_in_note	X	X	X	X	
type_of_notation	X	X	X	X	
number_with_units	X	X	X	X	
units_of	X	X	X	X	
value_of	X	X	X	X	
other_reference_document	X	X	X	X	
part_occurrence_in_assembly			X	X	
location			X	X	
parts_list	X	X	X	X	
list_body	X	X	X	X	
list_header	X	X	X	X	
parts_list_body	X	X	X	X	
list_notes		X	X	X	
parts_tabulations	X	X	X	X	
referenced_tdp_elements		X	X	X	For drawing
revision_history	X	X	X	X	
parts_list_header	X	X	X	X	
common_header	X	X	X	X	
person				X	
first_name				X	
last_name				X	
middle_name				X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
title				X	
unique_identifier				X	
person_and_organization				X	
electronic_mail_address				X	
organization				X	
person_address				X	
person_identification				X	
product_configuration	X	X	X	X	
approval_of_configuration				X	
assigned_to				X	
configuration_id	X	X	X	X	
design_item			X	X	
product_name	X	X	X	X	
product_data_set				X	
file_configuration				X	
related				X	
product_data_set_with_format				X	
presentation_of_product_data_set				X	
product_header				X	
product_data_set_with_shading				X	
a_shaded_model				X	
data_configuration				X	
product_data_set_without_format				X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
a_model				X	
data_configuration				X	
product_model	X	X	X	X	
model_name	X	X	X	X	
projects				X	
project				X	
description				X	
end_date				X	
existence				X	
id				X	
name				X	
participants				X	
project_owner				X	
start_date				X	
promissory_usage			X	X	
property_assignment				X	
described_element				X	
description				X	
name				X	
quantified_part_usage_in_assembly			X	X	
quantity	X	X	X	X	
quantity_accuracy	X	X	X	X	
value_of	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
reference_document	X	X	X	X	
document_identification	X	X	X	X	
reference_document_usage			X	X	For material capability
document_referenced			X	X	For material capability
location_in_document			X	X	For material capability
usage_description			X	X	For material capability
usage_code			X	X	For material capability
usage_parameter			X	X	
relative_event				X	
offset				X	
related_event				X	
release_authentication	X	X	X	X	
authentication	X	X	X	X	
person_responsible				X	
release_authority				X	
release_authority_code				X	
release_date	X	X	X	X	
retrofit_usage				X	
defining_document				X	
disposition_for				X	
retrofit_description				X	
revision	X	X	X	X	
authorizing_documents	X	X	X	X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
revision_approval				X	
revision_date	X	X	X	X	
revision_description	X	X	X	X	
revision_level	X	X	X	X	
security_classification				X	
classification_date				X	
classifier				X	
declassification_date				X	
item_classification				X	
title_security_classification				X	
sequence_effectivity			X	X	For limited history
from_effectivity_id			X	X	For limited history
quantity_unit_of_measure			X	X	For limited history
thru_effectivity_id			X	X	For limited history
total_component_quantity_in_-product_configuration			X	X	
sheet				X	
presentation_format				X	
sheet_configuration				X	
sheet_identifier				X	
sheet_size				X	
size_characteristics				X	
byte_size				X	
sheet_size				X	

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
size_characteristics_full_size				X	
source_information_type				X	
source_code				X	
source_description				X	
type_of_coding				X	
special_condition		X	X	X	
code		X	X	X	
description				X	
type_of_coding_scheme		X	X	X	
specification_document			X	X	For material capability
specified_part_in_assembly_tree			X	X	
higher_assembly			X	X	
sub_assembly			X	X	
standard_document			X	X	For material capability
status			X	X	
status_code				X	
status_code_basis				X	
stock_material			X	X	For material capability
material			X	X	For material capability
size			X	X	For material capability
stock_size			X	X	For material capability
catalog			X	X	For material capability
cross_section			X	X	For material capability

Table K2.2 — Levels of capability for PL application objects and attributes (continued)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
parameters_for			X	X	For material capability
stock_size_parameters			X	X	For material capability
a_length			X	X	For material capability
diameter			X	X	For material capability
height			X	X	For material capability
name_of_parameter			X	X	For material capability
numeric_parameter			X	X	For material capability
radius			X	X	For material capability
textual_parameter			X	X	For material capability
wall_thickness			X	X	For material capability
width			X	X	For material capability
tdp_element	X	X	X	X	
tdp_element_and_item_association	X	X	X	X	
related_item				X	
related_tdp_element				X	
tdp_element_list_item		X	X	X	For drawing
reference_code		X	X	X	For drawing
tdp_element_item		X	X	X	For drawing
text	X	X	X	X	
time	X	X	X	X	
hour	X	X	X	X	
minute	X	X	X	X	
second	X	X	X	X	

Table K.2.2 — Levels of capability for PL application objects and attributes (concluded)

Parts list objects	Levels of capability				Notes
	1	2	3	4	
zone	X	X	X	X	
time_interval_effectivity				X	
duration				X	
interval_name				X	
primary_bound				X	
secondary_bound				X	
transformation				X	
volume	X	X	X	X	
a_value	X	X	X	X	
weight	X	X	X	X	
a_value				X	
derivation_method				X	
unit_of_measure	X	X	X	X	

K.3 Definitions and levels of capability for the data list

This appendix section defines the Data List (DL) and its associated information. The levels of capability for the DL focus on the ability to capture document management and accountability information for the purposes of exchanging the DL information between two enterprises or between two systems. These levels of capability do not address graphical presentation of the Data List.

K.3.1 Data list definition

The Data List is a tabulation of all engineering drawings, product data sets, associated lists, specifications, standards, and subordinate data lists pertaining to the item to which the data list applies and essential in-house documents necessary to meet the technical design disclosure requirements, except for those in-house documents referenced parenthetically.

Data captured on the Data List is comprised primarily of summary information about TDP elements or documents that include element identification (for example, document number, title), security, distribution

controls, and other pertinent information required to support the management of the TDP element. The Data List tabulation identifies those TDP elements that are called out for the item to which the data list applies and does not give a top-down break-down listing of all TDP elements below the level of the item.

K.3.2 Data list levels of capability

The functional utility of the DL is dependent on the amount of TDP element information recorded. The following levels of capability define an increasing level of utility attributed to the number of application objects used to comprise it.

K.3.2.1 Minimal level

This level provides a minimum level of document management and document accounting capability for commercial exchanges. The DL constructs identified allow for simple TDP element relationships and characteristics to be exchanged. Only mandatory application objects are identified, all other application objects in the following levels of capability are considered optional in a commercial exchange.

— This level of capability allows for the capture and exchange of the following information:

— identifies the DL document number and version;

— identifies the DL version and number of sheets;

— identifies tabulated documents and TDP elements by number, version, and nomenclature (such as, name);

— identifies for each tabulated document the drawing size (when applicable) and the number of sheets (for a drawing).

K.3.2.2 Controlled level

This level provides a higher level than the minimum level of document management and document accounting capability. Government requirements are usually more stringent than the commercial requirements in the definition of the developing design authority information and in configuration procedures for the DL.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities identified in minimum level above;

— identifies the contract number under which the DL was prepared;

— identifies the title of the DL;

— identifies the preparing activity of the DL and the preparing activity of the tabulated entries in the DL;

— identifies the release authentication (when required).

K.3.2.3 Supplemental identification level

The supplemental identification level of capability provides capabilities to those previously cited of identifying and managing TDP element using information which specifically characterizes the TDP element for data management and accountability functions.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities in the controlled level of capability;

— notation for each TDP elements, including special conditions identified for the TDP element;

— end item identification usage of the item that the DL was prepared for;

— security, distribution, and data rights information for the DL of interest and the tabulations on the DL.

K.3.3 Data list requirements traceability

Table K.3.1 and Table K.3.2 identify the application objects for the Data List (DL) and to which level of capability they apply. The columns in the matrix are correlated to the levels of capability as follows:

— Column 1 is the application object, enumeration, or select;

— Column 2 = Level 1 capability = Minimum level (commercial);

— Column 3 = Level 2 capability = Controlled level (government);

— Column 4 = Level 3 capability = Supplemental identification level;

— Column 5 contains notes of interest to the other columns.

Table K.3.1 defines the levels of capability for DL EXPRESS constructs for selects and enumerations. Table K.3.2 defines the levels of capability for DL applications objects.

Table K.3.1 — Levels of capability for DL selects and enumerations

Data list Selects and enumerations	Levels of capability			Notes
	1	2	3	
Enumerations				
change_type			X	

Table K.3.1 — Levels of capability for DL selects and enumerations (concluded)

Data list Selects and enumerations	Levels of capability			Notes
	1	2	3	
Selects				
alternate_identification_element_select			X	
alternate_identification_item_select			X	
drawing_or_product_data_set			X	
item_identifying_number_select			X	
revision_authorization_select	X	X	X	
source_identification			X	
valid_tdp_elements_for_data_list	X	X	X	

Table K.3.2 — Levels of capability for DL application objects and attributes

Data list objects	Levels of capability			Notes
	1	2	3	
a_number	X	X	X	
a_real	X	X	X	
alternate_element_identification			X	
change_status			X	
design_activity			X	
element_classifications			X	
identifying_number			X	
outstanding_changes			X	
title			X	
alternate_identification_of_item			X	
change_status			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
classifications			X	
design_activity			X	
identifying_number			X	
nomenclaature_or_name			X	
source_information			X	
an_integer	X	X	X	
approval			X	
approval_date			X	
person_organization			X	
purpose_of			X	
status			X	
associated_list	X	X	X	
presentation				This is for presentation of the information only.
related_to			X	
certification			X	
approvals			X	
creation_date			X	
description			X	
name			X	
change_identification	X	X	X	
action_status			X	
assigned_to			X	
change_code			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
change_date			X	
change_description			X	
change_level			X	
issue_date			X	
issue_level			X	
revision_authorization_identifications	X	X	X	
revision_date	X	X	X	
revision_description	X	X	X	
revision_level	X	X	X	
company	X	X	X	
company_code		X	X	
code		X	X	
code_administrator		X	X	
type_of_code		X	X	
configuration	X	X	X	
data_usage_rights			X	
distribution_authorization			X	
end_item_system_designation			X	
preparing_contracts		X	X	
release_authorizations	X	X	X	
security_identifications			X	
contract		X	X	
affected_organizations			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
approvals			X	
contract_data_requirements_list			X	
contract_number		X	X	
creation_date			X	
date_item_description_identification			X	
contract_submission			X	
date_of_submission			X	
delivered_contract			X	
location			X	
data_list	X	X	X	
body	X	X	X	
list_header	X	X	X	
data_list_body	X	X	X	
data_list_notes			X	
data_list_tabulations	X	X	X	
revision_history	X	X	X	
data_list_entry	X	X	X	
available_from			X	
entry_item_change_level			X	
entry_notes			X	
list_entry	X	X	X	
special_conditions			X	
data_list_header	X	X	X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
common_header	X	X	X	
data_list_tabulation	X	X	X	
data_list_entries	X	X	X	
for_item			X	
date	X	X	X	
day	X	X	X	
month	X	X	X	
specific_time	X	X	X	
week	X	X	X	
year	X	X	X	
design_authority	X	X	X	
design_activity_code		X	X	
design_activity_identification	X	X	X	
person_responsible			X	
role			X	
distribution_notice			X	
distribution_authority			X	
distribution_code			X	
distribution_statement			X	
document_list	X	X	X	
document_usage_parameter			X	
subject			X	
value_of				

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
drawing	X	X	X	For DL Tabulations
heading	X	X	X	For DL Tabulations
pages	X	X	X	For DL Tabulations
drawing_suffix_number_combination	X	X	X	
drawing_number	X	X	X	
suffix_identifier	X	X	X	if required
element_identification	X	X	X	
alternate_identifications	X	X	X	
change_status	X	X	X	if applicable
design_activities	X	X	X	
element_certification			X	
element_classifications		X	X	
identifying_number	X	X	X	
outstanding_changes			X	
title		X	X	
element_type		X	X	
code_administrator		X	X	
element_code		X	X	
type_of_coding_scheme		X	X	
header	X	X	X	
document_abstract		X	X	
document_keywords		X	X	
header_configuration	X	X	X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
language				
sheet_count	X	X	X	
size_of_sheet				
header_configuration_with_element_-identification	X	X	X	
data_configuration	X	X	X	
identification	X	X	X	
identifier	X	X	X	
item			X	
alternates				(N/A)
conditions_defined_through_-constrained_document				
conditions_defined_through_simple_-reference				
identification			X	
item_configuration			X	
item_contexts				
item_weight				(N/A)
notes			X	
shape				(N/A)
shape_aspects				(N/A)
special_conditions			X	
item_identification			X	
alternate_identifications			X	
change_status			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
classifications			X	
design_activity			X	
identifying_number			X	
item_certification			X	
nomenclature_or_name			X	
source_information			X	
item_type			X	
code_administrator			X	
item_code			X	
type_of_coding_scheme			X	
label	X	X	X	
notation			X	
note			X	
note_parameter			X	
note_title			X	
reference_code			X	
referenced_document_in_note			X	
type_of_notation			X	
other_list	X	X	X	For DL Tabulations
list_body				This is not to be used
list_header	X	X	X	For DL Tabulations
other_list_header	X	X	X	For DL Tabulations
common_header	X	X	X	For DL Tabulations

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
other_reference_document	X	X	X	
parts_list	X	X	X	For DL Tabulations
list_body				This is not to be used
list_header	X	X	X	For DL Tabulations
parts_list_header	X	X	X	For DL Tabulations
common_header	X	X	X	For DL Tabulations
person			X	
first_name			X	
last_name			X	
middle_name			X	
title			X	
unique_identifier			X	
person_and_organization			X	
electronic_mail_address			X	
organization			X	
person_address			X	
person_identification			X	
product_configuration			X	
approval_of_configuration			X	
assigned_to			X	
configuration_id			X	
design_item			X	
product_name			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
product_data_set	X	X	X	For DL Tabulations
file_configuration				This is not to be used
related			X	
product_data_set_with_format	X	X	X	For DL Tabulations
presentation_of_product_data_set				This is not to be used
product_header	X	X	X	For DL Tabulations
product_data_set_with_shading	X	X	X	
a_shaded_model				This is not used
data_configuration	X	X	X	For DL Tabulations
product_data_set_without_format	X	X	X	For DL Tabulations
data_configuration	X	X	X	For DL Tabulations
a_model				This is not used
product_model			X	
model_name			X	
projects				This is not used
reference_document	X	X	X	
document_identification	X	X	X	
reference_document_usage			X	
document_referenced			X	
location_in_document			X	
usage_description			X	
usage_code			X	
usage_parameter			X	

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
release_authentication	X	X	X	
authentication	X	X	X	
person_responsible			X	
release_authority			X	
release_authority_code			X	
release_date	X	X	X	
revision	X	X	X	
authorizing_documents	X	X	X	
revision_approval			X	
revision_date	X	X	X	
revision_description	X	X	X	
revision_level	X	X	X	
security_classification			X	
classification_date			X	
classifier			X	
declassification_date			X	
item_classification			X	
title_security_classification			X	
sheet	X	X	X	For DL Tabulations
presentation_format				Not to be used
sheet_configuration	X	X	X	For DL Tabulations
sheet_identifier	X	X	X	For DL Tabulations
sheet_size	X	X	X	For DL Tabulations

Table K.3.2 — Levels of capability for DL application objects and attributes (continued)

Data list objects	Levels of capability			Notes
	1	2	3	
single_document_list	X	X	X	
size_characteristics			X	
byte_size			X	
sheet_size			X	
size_characteristics_fule_size			X	
source_location			X	
path_information			X	
storage_node_identification			X	
special_condition			X	
code			X	
description			X	
type_of_coding_scheme			X	
specification_document	X	X	X	For DL Tabulations
standard_docmunent	X	X	X	For DL Tabulations
status			X	
status_code			X	
status_code_basis			X	
tdp_element	X	X	X	
tdp_element_and_item_association	X	X	X	
related_item	X	X	X	
related_tdp_element	X	X	X	
text	X	X	X	
time	X	X	X	

Table K.3.2 — Levels of capability for DL application objects and attributes (concluded)

Data list objects	Levels of capability			Notes
	1	2	3	
hour	X	X	X	
minute	X	X	X	
second	X	X	X	
zone	X	X	X	

K.4 Definitions and levels of capability for the index list

This appendix section defines the Index List (IL) and its associated information. The levels of capability for the IL focus on the ability to capture document management and accountability information for the purposes of exchanging the IL information between two enterprises or between two systems. These levels of capability do not address graphical presentation of the Index List.

K.4.1 Index list definition

The Index List is a tabulation of data lists and subordinate index lists pertaining to the item to which the index list applies.

Data captured on the Index List is comprised primarily of summary information about TDP elements or documents for purposes of collecting the TDP.

K.4.2 Index list levels of capability

The functional utility of the IL is dependent on the amount of TDP element information recorded. The following levels of capability define an increasing level of utility attributed to the number of application objects used to comprise it.

K.4.2.1 Minimal level

This level provides a minimum level of document management and document accounting capability for commercial exchanges. The IL constructs identified allow for simple TDP element relationships and characteristics to be exchanged. Only mandatory application objects are identified, all other application objects in the following levels of capability are considered optional in a commercial exchange.

This level of capability allows for the capture and exchange of the following information:

— identifies the IL document number and version;

— identifies the IL version and number of sheets;

— identifies tabulated TDP elements by number, version, and nomenclature (such as, name).

K.4.2.2 Controlled level

This level provides a higher level than the minimum level of document management and document accounting capability. Government requirements are more stringent than the commercial requirements in the definition of the developing design authority information and in configuration procedures for the IL.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities identified in minimum level above;

— identifies the contract number under which the IL was prepared;

— identifies the title of the IL;

— identifies the preparing activity of the IL and the preparing activity of the tabulated entries in the IL;

— identifies the release authentication (when required).

K.4.3 Supplemental identification level

The supplemental identification level of capability provides capabilities to those previously cited of identifying and managing TDP element using information which specifically characterizes the TDP element for data management and accountability functions.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities in the controlled level of capability;

— notation for each TDP elements, including special conditions identified for the TDP element;

— end item identification usage of the item that the IL was prepared for;

— security, distribution, and data rights information for the IL of interest and the tabulations on the IL.

K.4.4 Index list requirements traceability

Table K.4.1 and Table K.4.2 identify the application objects for the Index List (IL) and to which level of capability they apply. The columns in the matrix are correlated to the levels of capability as follows:

— Column 1 contains the application object, enumeration, or select;

— Column 2 = Level 1 = Minimum level (commercial);

— Column 3 = Level 2 = Controlled level (government);

— Column 4 = Level 3 = Supplemental identification level;

— Column 5 contains the notes of interest to the other columns.

Table K.4.1 defines the levels of capability for IL EXPRESS constructs for selects and enumerations. Table K.4.2 defines the levels of capability for IL applications objects.

Table K.4.1 — Levels of capability for IL selects and enumerations

Index list Selects and enumerations	Levels of capability			Notes
	1	2	3	
Enumerations				
change_type			X	
Selects				
alternate_identification_element_select				
alternate_identification_item_select				
drawing_or_product_data_set			X	
revision_authorization_select	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes

Index list objects	Levels of capability			Notes
	1	2	3	
a_number	X	X	X	
a_real	X	X	X	
alternate_element_identification			X	
change_status			X	
design_activity			X	
element_classifications			X	
identifying_number			X	
outstanding_changes			X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
title			X	
an_integer	X	X	X	
approval			X	
approval_date			X	
person_organization			X	
purpose_of			X	
status			X	
associated_list	X	X	X	
presentation				This is for presentation of the information only.
related_to			X	
certification			X	
approvals			X	
creation_date			X	
description			X	
name			X	
change_identification	X	X	X	
action_status			X	
assigned_to			X	
change_code			X	
change_date			X	
change_description			X	
change_level			X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
issue_date			X	
issue_level			X	
revision_authorization_identifications	X	X	X	
revision_date	X	X	X	
revision_description	X	X	X	
revision_level	X	X	X	
company	X	X	X	
company_code		X	X	
code		X	X	
code_administrator		X	X	
type_of_code		X	X	
configuration	X	X	X	
data_usage_rights			X	
distribution_authorizations			X	
end_item_system_designation			X	
preparing_contracts		X	X	
release_authorizations	X	X	X	
security_identifications			X	
contract		X	X	
affected_organizations			X	
approvals			X	
contract_data_requirments_list			X	
contract_number		X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
creation_date			X	
data_item_description_identification			X	
data_list	X	X	X	For IL Tabulations
body				Not to be used
list_header	X	X	X	For IL Tabulations
data_list_header	X	X	X	For IL Tabulations
common_header	X	X	X	For IL Tabulations
date	X	X	X	
day	X	X	X	
month	X	X	X	
specific_time	X	X	X	
week	X	X	X	
year	X	X	X	
design_authority	X	X	X	
design_activity_code			X	
design_activity_identification	X	X	X	
person_responsible			X	
role			X	
distribution_notice			X	
distribution_authority			X	
distribution_code			X	
distribution_statement			X	
document_list	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
drawing			X	
heading			X	
pages			X	
element_identification	X	X	X	
alternate_identifications	X	X	X	
change_status	X	X	X	if applicable
design_activity	X	X	X	
element_certification			X	
element_classifications		X	X	
identifying_number	X	X	X	
outstanding_changes			X	
title		X	X	
element_type		X	X	
code_administrator		X	X	
element_code		X	X	
type_of_coding_scheme		X	X	
header	X	X	X	
document_abstract		X	X	
document_keywords		X	X	
header_configuration	X	X	X	
language			X	
sheet_count	X	X	X	
size_of_sheet	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
header_configuration_with_element_-identification	X	X	X	
data_configuration	X	X	X	
identification	X	X	X	
identifier	X	X	X	
indentured_data_list	X	X	X	For IL tabulations
body				Not to be used
header	X	X	X	For IL tabulations
indentured_data_list_header	X	X	X	For IL tabulations
common_header	X	X	X	For IL tabulations
procurement_references			X	
index_list	X	X	X	
body	X	X	X	
list_header	X	X	X	
index_list_body	X	X	X	
index_list_entries	X	X	X	
index_list_notes			X	
revision_history	X	X	X	
index_list_entry	X	X	X	
entry_item_change_level			X	
entry_notes			X	
list_entry	X	X	X	
index_list_header	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
common_header	X	X	X	
item			X	
alternates				(N/A)
conditions_defined_through_-constrained_-document			X	
conditions_defined_through_simple_-reference			X	
identification			X	
item_configuration			X	
item_contexts			X	
item_weight				(N/A)
notes			X	
shape				(N/A)
shape_aspects				(N/A)
special_conditions			X	
item_identification			X	
alternate_identifications			X	
change_status			X	
classifications			X	
design_activity			X	
identifying_number			X	
item_certification			X	
nomenclature_or_name			X	
source_information			X	
label	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
notation			X	
note			X	
note_parameter			X	
note_title			X	
reference_code			X	
referenced_document_in_note			X	
type_of_notation			X	
other_reference_document	X	X	X	
person			X	
first_name			X	
last_name			X	
middle_name			X	
title			X	
unique_identifier			X	
person_and_organization			X	
electronic_mail_address			X	
organization			X	
person_address			X	
person_identification			X	
product_configuration			X	
approval_of_configuration			X	
assigned_to			X	
configuration_id			X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
design_item			X	
product_name			X	
product_model			X	
model_name			X	
projects				This is not used
reference_document	X	X	X	
document_identification	X	X	X	
reference_document_usage			X	
document_referenced			X	
location_in_document			X	
usage_description			X	
usage_code			X	
usage_parameter			X	
release_authentication	X	X	X	
authentication	X	X	X	
person_responsible			X	
release_authority			X	
release_authority_code			X	
release_date	X	X	X	
revision	X	X	X	
authorizing_documents	X	X	X	
revision_approval			X	
revision_date	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (continued)

Index list objects	Levels of capability			Notes
	1	2	3	
revision_description	X	X	X	
revision_level	X	X	X	
security_classification			X	
classification_date			X	
classifier			X	
declassification_date			X	
item_classification			X	
title_security_classification			X	
single_document_list	X	X	X	
size_characteristics			X	
byte_size			X	
sheet_size			X	
size_characteristics_full_size			X	
status			X	
status_code			X	
status_code_basis			X	
tdp_element	X	X	X	
tdp_element_and_item_association			X	
related_item			X	
related_tdp_element			X	
text	X	X	X	
time	X	X	X	
hour	X	X	X	

Table K.4.2 — Levels of capability for IL application objects and attributes (concluded)

Index list objects	Levels of capability			Notes
	1	2	3	
minute	X	X	X	
second	X	X	X	
zone	X	X	X	

K.5 Definitions and levels of capability for the Data Definition Exchange (DDE)

K.5.1 Data definition exchange

This appendix section defines the Data Definition Exchange and its associated information. The levels of capability for the DDE focus on the ability to capture the packaging/shipping information required to identify and relate among one another the product data exchanged, in whole or in part, between two enterprises or between two systems. Within the DDE focus, the ability to capture product and document management and accountability information is also included.

K.5.1.1 Definition

The Data Definition Exchange (DDE) is an intelligent list identifying Technical Data Package (TDP) Elements that potentially are exchanged, such as engineering drawings, associated lists, referenced documents, or any other data sets or documents related to product items. The DDE may be use to identify and record the total number of TDP Elements required to support the given product item. In addition, TDP Elements for one or more products, assemblies, or details can be identified together within a DDE. The DDE content can be as basic as a simple shipping list or as complex as a full inventory of design disclosure defining which TDP Elements are actually being exchanged. A full inventory of design disclosure provides such things as TDP element to TDP element relationships, TDP element to data file relationships, file format data, and delivery accounting information. The DDE may use an indentured hierarchical structure that identifies how elements are related. The DDE may also be exchanged without accompanying TDP elements.

K.5.1.2 Functional capture of data relationships

The utilization of data relationships within the DDE is one of the primary aspects for distinguishing the level of capabilities that are supported by the DDE. First, utilizing a simple list of TDP Elements being exchanged provides support for a minimum set of exchange application capabilities. Next, providing an indentured list relationship of TDP Elements supports a more rich set of application capabilities.

The basis of indenture for a DDE may be of three types. The first is a TDP Element to TDP Element (or document to document) relationship. For example, a drawing and all of its referenced documents are identified starting with the end item or assembly drawing and identifying all referenced drawings, associated lists, product data sets and other referenced documents. The second is a part number to part number hierarchical indenture list with intermingled subordinate indentures of part to TDP element relationships and TDP element to TDP element relationships. This provides the ability to specifically identify all TDP elements required to define each part or assembly. The third is similar to the second in that a part to part relationship is derivable, however, the indenture is structured to first define a document/drawing, then its disclosed or referenced parts, then for each part the subassembly documents/drawings disclosing the subassembly parts, and then finally for each subassembly document/drawing the subassembly component parts are defined. Thus the third indenture method provides a part based indenture method while capturing the subordinate indenture relationships of a TDP element to a part.

K.5.1.3 Functional capture of required data

Data captured in the DDE is comprised primarily of description information about the TDP elements including element identification (such as document number, title), security and distribution controls, file formats, delivery accounting, exchange existence knowledge and other pertinent information required to support the exchange of TDP elements.

K.5.1.4 Levels of capability

The functional utility of the DDE is dependent on the amount of TDP element information recorded. The following levels of capability define an increasing level of utility attributed to the quantity of information construct types valid for a particular level.

K.5.1.4.1 Shipping list

Provides a minimum level of exchange information for a set of TDP elements. The exchange information at a minimum can only consist of a simple list of TDP elements (documents) or a simple list of files being exchanged. This level of capability does NOT provide for relationship information between two or more TDP elements to be captured. This shipping list capability could be utilized for any collection of files, even related technical data files that are not typically included as design discloser information, that needed to be exchanged as part of a package within or among business enterprises.

This level of capability allows for the capture and exchange of the following information:

— identifies DDE and related product item(s);

— defines DDE version data including original/revision data;

— date that the DDE is created;

— simple list of TDP elements or data files that comprise TDP elements;

— allows for identification of TDP elements by number and version.

K.5.1.4.2 General packaging

Provides a basic level of exchange information for a set of TDP elements. In addition to the requirements for a shipping list being covered, this general packaging capability provides: TDP element to their corresponding data file relationships, data file digital format identification, and general TDP element description data such as security, source, and destination information. Also the capability is supported where two or more formats (digital or non-digital) are required for a single TDP element, thus enabling the receiving enterprise/system to identify, manage, and account for all distinct TDP element forms necessary to support the product item, assembly, or detail. An example would be a parts list TDP element with a computer processable form contained in a STEP format and a presentation form in a raster format.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities identified in Shipping List capability;

— identifies the different forms available for each TDP element (for example, raster, STEP, SGML, paper);

— provides TDP element to data file relationships;

— provides digital format of the data files;

— identifies total count of internal TDP element divisions and their division types such as the sheet count or pages;

— identifies document usage controls (unlimited or limited rights) for the DDE and TDP elements;

— provides system origination with release authorization and target system destination;

— identifies security level of document or title;

— date of transfer of the DDE;

— allows for the identification of TDP elements by document type and preparing design activity.

K.5.1.4.3 Indenture list

Provide relationships among TDP elements, among Product Items, and among TDP elements and Product Items. These relationships establish the capability to tie/merge product data management structure to document based management structures. There are three types of indenture for a DDE. Each indenture type is described in the following three subsections: K.5.1.4.3.1, K.5.1.4.3.2, K.5.1.4.3.3. The capability each indenture list type provides is not additive to each other but is applied independently to the other DDE capability levels. There are four types of parent child relationship that are utilized to formulate the three types of indenture lists. Table K.5.1 shows which parent child relationships are used in the three types of indenture lists.

Table K.5.1 — Parent child relationships used in indenture list types

INDENTURE LIST TYPE	PARENT/ CHILD			
	PART/ PART	PART/ TDP Element	TDP Element/ PART	TDP Element/ TDP Element
Indenture by TDP Element				Yes *
Indenture by Part	Yes *	Yes		Yes
Indenture by Part (with Document References to Parts)	Yes *	Yes *	Yes *	Yes *

Notes: * - Top Indenture

K.5.1.4.3.1 Indenture by TDP element

Provides the ability to capture the relationships of TDP elements within the exchange. A hierarchical indenture list that provides a parent child relationship between the TDP elements is supported. Applications can query this list to identify what documents are referenced from any other. For example, the relationship between an assembly drawing and its subordinate documents (such as detail drawings, material specification) may be captured.

This level of capability allows for the capture and exchange of the following information:

— identifies parent child relationships between TDP elements;

— allows for a hierarchical order of indenture to be captured among TDP elements.

K.5.1.4.3.2 Indenture by part

Provides the ability to capture the relationships of parts within the exchange. A hierarchical indentured list that provides a parent child relationship between parts and assemblies, parts and their TDP elements, and TDP elements and their subordinate TDP elements is supported. Applications may query this list to identify what documents are available about a part, what parts go into which assembly, and what documents are referenced from other documents. For example, utilizing the Indenture by Part capability level, the drawing a part is defined in and the assembly that a part is a portion of could be determined by just knowing the identification of a part.

This level of capability allows for the capture and exchange of the following information:

— identifies parent child relationships between parts;

— identifies parent child relationships between a part (parent) and a TDP Element (child);

— identifies parent child relationships between TDP elements;

— allows for a hierarchical order of indenture to be captured among parts and TDP elements with the part always playing the higher indenture role when associated directly with a TDP Element;

— identifies that a part always occupies the first indenture level.

K.5.1.4.3.3 Indenture by part (with document references to parts)

Provides the ability to capture the relationships of parts and/or TDP Elements within the exchange. A hierarchical indentured list that provides a parent child relationship between parts and assemblies, parts and their TDP elements, TDP elements and their parts, and TDP elements and their subordinate TDP elements is supported. Applications may query this list to identify what documents are available about a part, what parts go into which assembly, what parts are defined in a drawing, and what documents are referenced from other documents. For example, utilizing the Indenture by Part (with Document References to Parts) capability level, the parts defined on an assembly drawing could be identified by just knowing the identification of the assembly drawing.

This level of capability allows for the capture and exchange of the following information:

— identifies an implied parent child relationships between parts;

— identifies parent child relationships between a part (parent) and a TDP Element (child);

— identifies parent child relationships between a TDP Element (parent) and a part (child);

— identifies parent child relationships between TDP elements;

— allows for a hierarchical order of indenture to be captured among parts and TDP elements with either the part or the TDP Element playing the higher indenture role;

— allows that just the TDP_element occupy the first indenture level.

K.5.1.4.3.4 Delivery accounting

Provides the ability to capture tracking parameters so that traceability of shipments may be performed. Identification of formal shipping documents/forms such as Letter of Transmittals or U.S.A. government form DD250 may be captured to provide a relationship with legacy delivery accounting systems. Delivery Accounting information may be captured not only for the specific information that makes up the DDE itself but also for any TDP Element entry with in the DDE.

This level of capability allows for the capture and exchange of the following information:

— allows for the identification of Shipping List identification;

— allows for the identification of Purchase Orders;

— provides for the identification of associated Letters of Transmittals;

— provides for the identification of related Material Receiving Inspection Forms;

— reason for the exchange may be captured;

— identifies the procurement control drawing(s) (PCD) or specification control drawing (SCD) for which the DDE is developed;

— identifies the contract under which the TDP element was developed if not developed specifically for the contract authorizing the DDE.

K.5.1.4.3.5 Supplemental identification parameters

Provides additional capabilities to those previously cited of identifying and managing TDP elements using information which specifically characterizes the TDP elements for data management and accountability functions. This type of information includes data access knowledge, size characteristics, plus some general entry note and change information.

This level of capability allows for the capture and exchange of the following information:

— identification of size_characteristic such as byte size for files and sheet size for layout or physical presentation;

— defines data access knowledge for remote or direct access to TDP data files such as path information or storage node identification;

— allows for the identification status of the life cycle and the DDE and TDP element entry are in during the exchange;

— allows for the identification of note and change information of each tdp element entry;

— identifies the serialization (effectivity) of the listed part.

K.5.1.5 Relation to other techncial data elements

The concept of the DDE is similar to a shipping list that is used to identify the documents or product data being exchanged. However, the DDE also represents all of the same information available in the IDL. The two dominant characteristics differentiating the two concepts (that is, the DDE, and IDL) is that the DDE is not intended to take on the form of a physical document, and the IDL does constitute a design disclosure element for which it is its primary purpose.

The DDE can be used in conjunction with the IDL. The DDE constitutes a single source of document exchange metadata as an alternative or complementary to other digital exchange technologies (for example, EDIFACT[8], ANSI X.12/EDI[7], United States MIL-STD-1840[9], or FTP).

K.5.2 Levels of capability resource matrix

The following matrix defines Data Definition Exchange (DDE) objects and to which level of capability they apply. The columns in the matrix are correlated to the levels of capability as follows:

— Column 1 - Column 1 is the application object, enumeration, or select;

— Column 2 = Level 1 = Shipping list;

— Column 3 = Level 2 = General packaging;

— Column 4 = Level 3 = Indentured list;

— Column 5 = Level 4 = Delivery accounting;

— Column 6 = Level 5 = Supplemental Identification Parameters.

The listing of DDE objects and their assigned levels of capability are defined in two tables. Table K.5.2 defines the levels of capability for DDE base types, selects, and enumeration objects. Table K.5.3 defines the levels of capability for DDE entities and their associated attributes.

Each column in Table K.5.2 and Table K.5.3 reflecting a capability level only shows additive capability that would be combined with the lower numbered capability levels.

Table K.5.2 — Levels of capability for DDE, selects, and enumerations

Data definition exchange SELECTS AND ENUMERATIONS	Levels of capability				
	1	2	3	4	5
ENUMERATIONS					
change_type	X				
change		X			
original	X				
original_with_change		X			
other		X			
revision	X				
revision_with_change		X			
exchange_reason		X			
quantity_accuracy_enumeration					

Table K.5.2 — Levels of capability for DDE, selects, and enumerations (continued)

Data definition exchange SELECTS AND ENUMERATIONS	Levels of capability				
	1	2	3	4	5
SELECTS					
alternate_identification_element_select					X
alternate_identification_item_select					X
data_definition_entry_select			X		
data_definition_entry_and_file_select			X		
data_definition_exchange_list_method	X				
data_definition_indentured_list_method		X			
simple_list_of_elements	X				
simple_list_of_files	X				
data_definition_file_entry		X			
accessed_file					X
exchange_file		X			
date_or_event					X
entry_format_select		X			
file_format		X			
text		X			
identification_select		X			
element_identification		X			
item_identification			X		
indentured_list_method			X		
indentured_list_by_document			X		
indentured_list_by_part				X	
indentured_list_by_part_with_document_-references_to_parts			X		
item_identifying_number_select			X		

Table K.5.2 — Levels of capability for DDE, selects, and enumerations (concluded)

Data definition exchange SELECTS AND ENUMERATIONS	Levels of capability				
	1	2	3	4	5
drawing_suffix_number_combination			X		
identifier			X		
item_or_element			X		
tdp_indentured_item			X		
tdp_indentured_tdp_element			X		
list_entry_select					
quantity_accuracy_select					X
quantity_accuracy_enumeration					X
text					X
quantity_type_select					X
a_number					X
text					X
volume					X
weight					X
revision_authroization_select	X				
identifier	X				
other_reference_document	X				
shape_select					X
source_identification					X
contract_submission					X
reference_document					X
usage_context					X

Table K.5.3 — Levels of capability for DDE application objects and attributes

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
a_number	X				
a_real	X				
accessed_file					X
path_information					X
storage_node_identification					X
alternate_element_identification					X
change_status					X
design_activity					X
element_classifications					X
identifying_number					X
outstanding_changes					X
title					X
alternate_identification_of_item					X
change_status					X
classifications					X
design_activity					X
identifying_number					X
nomenclature_or_name					X
source_information					X
alternate_item					X
interchange_item					X
preference_order					X
type_of_alternate					X
usage_conditions					X
an_integer	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
approval					X
approval_date					X
person_organization					X
purpose_of					X
status					X
certification					X
approvals					X
creation_date					X
description					X
name					X
change_identification	X				
action_status		X			
assigned_to		X			
change_code		X			
change_date		X			
change_description		X			
change_level		X			
issue_date		X			
issue_level		X			
revision_authorization_identifications	X				
revision_date	X				
revision_description	X				
revision_level	X				
company	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
company_code	X				
code	X				
code_administrator	X				
type_of_code	X				
configuration	X				
data_usage_rights		X			
distribution_authorizations		X			
end_item_system_designation		X			
preparing_contracts					X
release_authorizations	X				
security_identifications		X			
content_property					X
detail_level					X
geometry_type					X
languages					X
real_world_scale					X
contract	X				
affected_organizations				X	
approvals				X	
contract_data_requirements_list				X	
contract_number	X				
creation_date				X	
data_item_description_identification				X	
contract_submission					X
date_of_submission					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
delivered_contract					X
location					X
data_definition_entry_item				X	
available_from					X
delivery_accounting_references				X	
entry_files	X				
entry_item_change_level					X
entry_notes	X				
special_conditions					X
superseded_entry					X
data_definition_entry_tdp_element	X				
actual_inclusion_in_data_exchange				X	
available_from					X
data_usage_rights		X			
delivery_accounting_references				X	
entry_content_property					X
entry_files	X				
entry_format		X			
entry_item_change_level					X
entry_notes	X				
master_file					X
size					X
special_conditions					X
superseded_entry					X
data_definition_exchange	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
list_body	X				
list_header	X				
presentation	X				
data_definition_exchange_body	X				
list_method	X				
notes_list					X
revision_history					X
data_definition_exchange_header	X				
common_header	X				
date_of_transfer		X			
delivery_accounting_references				X	
destinations		X			
exchange_reason				X	
procurement_references					X
data_definition_exchange_simple_entry	X				
entry	X				
simple_entry	X				
data_definition_indentured_entry			X		
effective_on			X		
entry_characteristics			X		
indenture_level			X		
indentured_entry			X		
data_definition_indentured_list_method			X		
method_of_entry_tabulation			X		
order_of			X		

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
date	X				
day	X				
month	X				
specific_time	X				
week	X				
year	X				
date_effectivity					X
end_date					X
start_date					X
delivery_accounting				X	
reference_identification				X	
reference_identification_description				X	
design_authority		X			
design_activity_code		X			
design_activity_identification		X			
person_responsible					X
role					X
distribution_notice		X			
distribution_authority					X
distribution_code		X			
distribution_statement		X			
document_usage_parameter					X
subject					X
value_of					X
drawing_suffix_number_combination			X		

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
drawing_number			X		
suffix_identifier			X		
effectivity					X
description					X
effectivity_approvals					X
name					X
product					X
work_activity					X
element_identification	X				
alternate_identifications	X				
change_status	X				
design_activities	X				
element_certification					X
element_classifications					X
identifying_number	X				
outstanding_changes					X
title					X
element_type					X
code_administrator					X
element_code					X
type_of_coding_scheme					X
event					X
actual_date					X
assignment					X
description					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
id					X
name					X
planned_date					X
responsible_person_organization					X
exchange_file		X			
destination_system		X			
included_in_exchange					X
external_library_reference					X
description					X
external_id					X
library_type					X
file		X			
assigned_to					X
change_status					X
context_file_format		X			
context_file_name		X			
distribution_authorization		X			
file_content_property		X			
file_content_type		X			
file_note				X	
native_format_file_name					X
security_identifications		X			
size					X
source_system		X			
file_format		X			

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
change_level		X			
format_code		X			
format_standard		X			
release_date		X			
revision_level		X			
file_relationship			X		
description			X		
name			X		
related_file			X		
relating_file			X		
header	X				
document_abstract	X				
document_keywords	X				
header_configuration	X				
language	X				
sheet_count	X				
size_of_sheet	X				
header_configuration_with_element_identification	X				
data_configuration	X				
identification	X				
identifier	X				
indentured_list_by_document			X		
top_indentured_tdp_elements			X		
indentured_list_by_part			X		
top_indentured_items			X		

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
indentured_list_by_part_with_document_-references_to_parts			X		
top_indentured			X		
independent_property					X
allowed_units					X
description					X
id					X
property_source					X
independent_property_relationship					X
description					X
related_property					X
relating_property					X
relation_type					X
independent_property_usage					X
assigned_property					X
assigned_to					X
item			X		
alternates					X
conditions_defined_through_constrained_-document					X
conditions_defined_through_simple_reference					X
identification			X		
item_configuration					X
item_contexts					X
item_weight					X
notes					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
shape					X
shape_aspects					X
special_conditions					X
item_identification			X		
alternate_identifications			X		
change_status			X		
classifications			X		
desgin_activity			X		
identifying_number			X		
item_certification					X
nomenclature_or_name					X
source_information					X
item_parent_to_item_child_relationship			X		
child			X		
parent			X		
item_parent_to_tdp_element_child_relationship			X		
child			X		
parent			X		
item_type					X
code_administrator					X
item_code					X
type_of_coding_scheme					X
label	X				
lot_effectivity					X
lot_identification					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
lot_size					X
lot_unit_of_measure					X
measure					X
notation	X				
note	X				
note_parameter					X
note_title	X				
reference_code	X				
referenced_document_in_note					X
type_of_notation	X				
number_with_units					X
units_of					X
value_of					X
other_reference_document					X
person	X				
first_name	X				
last_name	X				
middle_name	X				
title	X				
unique_identifier	X				
person_and_organization	X				
electronic_mail_address	X				
organization	X				
person_address	X				
person_identification	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
product_configuration		X			
approval_of_configuration		X			
assigned_to					
configuration_id		X			
design_item					
product_name		X			
product_model		X			
model_name		X			
projects					X
project					X
description					X
end_date					X
existence					X
id					X
name					X
participats					X
project_owner					X
start_date					X
property_assignment					X
described_element					X
description					X
name					X
quantity					X
quantity_accuracy					X
value_of					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
reason				X	
applicable_to				X	
base				X	
exchange_purpose				X	
reference_document					X
document_identification					X
reference_document_usage					X
document_referenced					X
location_in_document					X
usage_description					X
usage_code					X
usage_parameter					X
relative_event					X
offset					X
related_event					X
release_authentication	X				
authentication	X				
person_responsible	X				
release_authority	X				
release_authority_code	X				
release_date	X				
revision	X				
authorizing_documents	X				
revision_approval					X
revision_date	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
revision_description					X
revision_level	X				
security_classification		X			
classification_date		X			
classifier		X			
declassification_date		X			
item_classification		X			
title_security_classification		X			
sequence_effectivity					X
from_effectivity_id					X
quantity_unit_of_measure					X
thru_effectivity_id					X
total_component_quantity_in_product_-configuration					X
sheet		X			
presentation_format					X
sheet_configuration					X
sheet_identifier		X			
sheet_size		X			
simple_list_of_elements	X				
element_entries	X				
simple_list_of_files	X				
file_entries	X				
size_characteristics		X			
byte_size					X

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
sheet_size		X			
size_characteristics_full_size		X			
size_characteristics_internal_divisions		X			
internal_division_count		X			
internal_division_type		X			
size_characteristics_sheet_across_file		X			
frame_number_comprising_this_file		X			
number_of_frames_for_sheet		X			
source_information_type					X
source_code					X
source_description					X
type_of_coding_scheme					X
source_location					X
path_information					X
storage_node_identification					X
special_condition					X
code					X
description					X
type_of_coding_schema					X
specification_document	X				
standard_document	X				
status					X
status_code					X
status_code_basis					X
system_declaration		X			

Table K.5.3 — Levels of capability for DDE, application objects and attributes (continued)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
creating_interface		X			
operating_system		X			
system		X			
system_element_identifier		X			
system_related_element_identifier		X			
system_destination		X			
target_company		X			
target_company_code_code		X			
target_person		X			
tdp_element	X				
tdp_element_and_item_association	X				
related_item	X				
related_tdp_element	X				
tdp_element_parent_to_item_child_relationship			X		
child			X		
parent			X		
tdp_element_parent_to_tdp_element_child_-relationship			X		
child			X		
parent			X		
tdp_indentured_item			X		
top_indenture			X		
tdp_indentured_tdp_element			X		
top_indenture			X		
text	X				

Table K.5.3 — Levels of capability for DDE, application objects and attributes (concluded)

Data definition exchange OBJECTS	Levels of capability				
	1	2	3	4	5
time	X				
hour	X				
minute	X				
second	X				
zone	X				
time_interval_effectivity					X
duration					X
interval_name					X
primary_bound					X
secondary_bound					X
volume					X
weight					X

K.6 Definitions and levels of capability for the indentured data list

K.6.1 Indentured data list

This appendix section defines the Indentured Data List (IDL) and its associated information. The levels of capability for the IDL focus on the ability to capture document management and accountability information for the purposes of exchanging this IDL information (and thus capability) between two enterprises or between two systems.

K.6.1.1 Definition

The Indentured Data List (IDL) is a design document disclosing a master list of Technical Data Package elements (TDP elements) such as engineering drawings, associated lists, referenced documents, or other design disclosure data which together provide product design data for a given product, assembly, or detail. The IDL defines the listing of TDP element by using an indentured hierarchical structure which identifies how elements are related. The IDL is used to identify and record the total number of TDP elements required to support the given product item.

K.6.1.2 Functional capture of data relationships

The basis of indenture for an IDL may be of three types. The first is a TDP element to TDP element (or document to document) relationship. For example, a drawing and all of its referenced documents are identified starting with the end item or assembly drawing and identifying all referenced drawings, associated lists, product data sets and other referenced documents. The second is a part number to part number hierarchical indenture list with intermingled subordinate indentures of part to TDP element relationships and TDP element to TDP element relationships. This provides the ability to specifically identify all TDP elements required to define each part or assembly. The third is similar to the second in that a part to part relationship is derivable, however, the indenture is structured to first define a document/drawing, then its disclosed or referenced parts, then for each part the subassembly documents/drawings disclosing the subassembly parts, and then finally for each subassembly document/drawing the subassembly component parts are defined. Thus the third indenture method provides a part based indenture method while capturing the subordinate indenture relationships of a TDP element to a part.

K.6.1.3 Functional capture of required data

Data captured on the IDL is comprised primarily of summary information about the TDP elements including related elements (parent or children), element identification (such as document number and title), security and distribution controls, and other pertinent information required to support the management of the element.

K.6.1.4 Levels of capability

The functional utility of the IDL is dependent on the amount of TDP element information recorded. The following levels of capability define an increasing level of utility attributed to the number of application objects used to comprise it. All levels of capability defined in the following sections utilized one of the three indentured methods described in section K.5.1.4.3.

K.6.1.4.1 Minimum level

Provides a minimum level of document management and document accounting capability for commercial exchanges. The IDL constructs identified allow for simple TDP element relationships and characteristics to be exchanged. Both mandatory and optional data are identified.

This level of capability allows for the capture and exchange of the following information:

— identifies the IDL document number and version;

— allows for identification of documents/TDP element by number and version;

— allows for identification of outstanding changes to document;

— defines element relationships by using a document indenture basis;

— defines the revision authorization document number;

— identifies general identification information (sheet count).

K.6.1.4.2 Controlled level

Provides a controlled level of document management and document accounting capability for government exchanges (such as, government or military contracts involving procurement, reprocurement, repair, maintenance, analysis)

The constructs identified encompass that capability captured by the commercial IDL plus additional mandatory information identifying contract, design activity, and other critical factors.

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities identified in the minimum level (Commercial);

— identifies the design activity responsible for the end item/IDL by identifying number and name;
— identifies the contract number under which the IDL was prepared;

— defines the title of the IDL;

— defines the end item or system designator for which the IDL was prepared;

— release_authentication;

— allows for identification of documents/TDP element by document type and preparing design activity;

— identifies document usage controls (unlimited or limited rights);

— identifies general application data (source/usage identification and special conditions).

K.6.1.4.3 Part number indenture basis

Provides a capability of managing and accounting for documents based on a designed component basis. A basic indenture method is a document-based indenture method. For a part indenture basis, a listing of part numbers to engineering document number (TDP element number) relationships and detail/subassembly to next higher assembly/end item TDP element relationships are captured for cross-reference use. The part number indenture basis may be captured in either of two ways. The first is part to part indenture relationship. The second is a document/drawing to part indentured relationship. The part indenture methods defined in this level of capability are described in the following two sections. The capability each indenture list type provides is not additive to each other but is applied independently to the other IDL capability levels.

Together, these three indenture methods (the original document-based indenture method, and the two part-based indenture methods defined for this level of capability) provide four types of parent-child relationships among TDP elements and Product Items. Table K.6.1 shows which parent child relationships are used in the three types of indenture lists.

Table K.6.1 — Parent child relationships used in indenture list types

INDENTURE LIST TYPE	PARENT/ CHILD			
	PART/ PART	PART/ TDP Element	TDP Element/ PART	TDP Element/ TDP Element
Indenture by TDP Element				Yes *
Indenture by Part	Yes *	Yes		Yes
Indenture by Part (with Document References to Parts)	Yes *	Yes *	Yes *	Yes *

NOTE * - Top Indenture

K.6.1.4.4 Indenture by part

Provides the ability to capture the relationships of parts within the exchange. A hierarchical indentured list that provides a parent child relationship between parts and assemblies, parts and their TDP elements, and TDP elements and their subordinate TDP elements is supported. Applications may query this list to identify what documents are available about a part, what parts go into which assembly, and what documents are referenced from other documents. For example, utilizing the Indenture by Part capability level, the drawing a part is defined in and the assembly that a part is a portion of could be determined by just knowing the identification of a part.

— This level of capability allows for the capture and exchange of the following information:

— identifies parent child relationships between parts;

— identifies parent child relationships between a part (parent) and a TDP Element (child);

— identifies parent child relationships between TDP elements;

— allows for a hierarchical order of indenture to be captured among parts and TDP elements with the part always playing the higher indenture role when associated directly with a TDP Element.;

— identifies that a part always occupies the first indenture level.

K.6.1.4.5 Indenture by part (with document references to parts)

Provides the ability to capture the relationships of parts and/or TDP Elements within the exchange. A hierarchical indentured list that provides a parent child relationship between, parts and assemblies, parts

and their TDP elements, TDP elements and their parts, and TDP elements and their subordinate TDP elements is supported. Applications may query this list to identify what documents are available about a part, what parts go into which assembly, what parts are defined on a drawing, and what documents are referenced from other documents. For example, utilizing the Indenture by Part (with Document References to Parts) capability level, the parts defined on an assembly drawing could be identified by just knowing the identification of the assembly drawing.

This level of capability allows for the capture and exchange of the following information:

— identifies an implied parent child relationships between parts;

— identifies parent child relationships between a part (parent) and a TDP Element (child);

— identifies parent child relationships between a TDP Element (parent) and a part (child);

— identifies parent child relationships between TDP elements;

— allows for a hierarchical order of indenture to be captured among parts and TDP elements with

— either the part or the TDP Element playing the higher indenture role;

— allows that just the TDP_element occupy the first indenture level.

K.6.1.4.6 Supplemental identification parameters

Provides additional capabilities to those previously cited of identifying and managing TDP element using information which specifically characterizes the TDP element for data management and accountability functions. This type of information includes national security classification, related IDL relationship data, related procurement or authorizing requirement information (for example, for tracing contractor to subcontractor IDL development).

This level of capability allows for the capture and exchange of the following information:

— all of the capabilities in any of the above capability levels;

— defines the security classification of the TDP element or the title of the TDP element;

— identifies the procurement control drawing(s) (PCD) or specification control drawing (SCD) for which the IDL is developed;

— identifies related IDL documents;

— defines the contract under which the TDP element was developed if not developed specifically for the contract authorizing the IDL;

— identifies the TDP element document as generally available through some identified service library or register (for example, the U.S. Department of Defense Index of Specifications and Standards).

K.6.1.5 Relation to other technical data elements

The concept of the IDL is a recent development founded from another technical data element called a Master Engineering Document List (MEDL). The IDL may be substituted for the MEDL when the part number indenture basis is used. The IDL is currently recognized as a design disclosure document.

The IDL is also functionally related to Data Lists and Index Lists and may be used to replace all data lists and an index list of all data lists for a given product item. In addition, the IDL may be used to identify a cross-reference for part numbers and their associated drawings and therefore maintains some Parts List functionality.

The IDL is also closely related to Data Definition Exchange (DDE) technical data element under the TDP AP development. The capability to identify all elements comprising full design disclosure for a given level of product, assembly, or detail, using an indentured structure is a common characteristic between these two elements. The primary differences are that 1) the DDE handles simple list of files or TDP_elements, and 2) the DDE is used primarily for facilitating exchanges.

The IDL could be used in conjunction with the DDE.

K.6.2 Requirements traceability

The following matrix defines Indentured Data List (IDL) objects and to which level of capability they apply. The columns in the matrix are correlated to the levels of capability as follows:

— Column 1 - Column 1 is the application object, enumeration, or select;

— Column 2 = Level 1 = Minimum Level;

— Column 3 = Level 2 = Controlled Level;

— Column 4 = Level 3 = Part Number Indenture Basis;

— Column 5 = Level 4 = Supplemental Identification Parameters;

The listing of IDL objects and their assigned levels of capability are defined in two tables. Table K.6.2 defines the levels of capability for IDL base types, selects, and enumeration objects. Table K.6.3 defines the levels of capability for IDL entities and their associated attributes.

Each column in Table K.6.2 and K.6.3 reflecting a capability level only shows additive capability that would be combined with the other capability levels.

Table K.6.2 — Levels of capability for IDL selects, and enumerations

Indentured data list SELECTS AND ENUMERATIONS	Levels of capability			
	1	2	3	4
ENUMERATIONS				
change_type	X			
original_with_change				X
revision	X			
revision_with_change				X
change				X
other				X
original	X			
quantity_accuracy_enumeration				X
approximate				X
as required				X
exact				X
SELECTS				
alternate_identification_element_select				X
alternate_identification_item_select				X
drawing_or_product_data_set				X
date_or_event				X
identification_select	X			
item_identification			X	
element_identification	X			
indentured_list_method	X			
indentured_list_by_part			X	
indentured_list_by_document	X			
indentured_list_by_part_with_document_-references_to_parts			X	
item_identifying_number_select			X	
identifier			X	

TABLE K.6.2 — Levels of capability for IDL selects and enumerations (continued)

Indentured data list SELECTS AND ENUMERATIONS	Levels of capability			
	1	2	3	4
drawing_suffix_number_combination			X	
item_or_element	X			
tdp_indentured_tdp_element	X			
tdp_indentured_item	X			
list_entry_select	X			
item_parent_to_item_child_relationship			X	
tdp_element_parent_to_item_child_relationship			X	
tdp_element_parent_to_tdp_element_child_ relationship	X			
item_parent_to_tdp_element_child_relationship			X	
tdp_indentured_tdp_element	X			
tdp_indentured_item			X	
quantity_accuracy_select				X
quantity_accuracy_enumeration				X
text				X
quantity_type_select				X
a_number				X
text				X
volume				X
weight				X
revision_authorization_select	X			
other_reference_document	X			
identifier	X			
shape_select				
source_identification				X

Table K.6.2 — Levels of capability for IDL selects and enumerations (concluded)

Indentured data list SELECTS AND ENUMERATIONS	Levels of capability			
	1	2	3	4
contract_submission				X
reference_document				X
usage_context				X

Table K.6.3 — Levels of capability for IDL application objects and attributes

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
a_number	X			
a_real	X			
alternate_element_identification				X
change_status				X
design_activity				X
element_classification				X
identifying_number				X
outstanding_changes				X
title				X
alternate_identification_of_item				X
change_status				X
classifications				X
design_activity				X
identifying_number				X
nomenclature_or_name				X
source_information				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
alternate_item				X
interchange_item				X
preference_order				X
type_of_alternate				X
usage_conditions				X
an_integer	X			
approval				X
approval_date				X
person_organization				X
purpose_of				X
status				X
associated_list	X			
presentation				X
related_to				X
certification				X
approvals				X
creation_date				X
description				X
name				X
change_identification	X			
action_status				X
assigned_to				X
change_code				X
change_date				X
change_description				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
change_level				X
issue_date				X
issue_level				X
revision_authorization_identifications	X			
revision_date	X			
revision_description	X			
revision_level	X			
company	X			
company_code	X			
code	X			
code_administrator	X			
type_of_code	X			
configuration	X			
data_usage_rights		X		
distribution_authorizations				X
end_item_system_designation	X			
preparing_contracts				X
release_authorizations	X			
security_authorizations				X
content_property				X
detail_level				X
geometry_type				X
languages				X
real_world_scale				X
contract		X		

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
affected_organizations				X
approvals				X
contract_data_requirements_list				X
contract_number		X		
creation_date				X
data_item_description_identification				X
contract_submission				X
date_of_submission				X
delivered_contract				X
location				X
date	X			
day	X			
month	X			
specific_time	X			
week	X			
year	X			
date_effectivity				X
end_date				X
start_date				X
design_authority		X		
design_activity_code		X		
design_activity_identification		X		
person_responsible				X
role				X
distribution_notice				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
distribution_authority				X
distribution_code				X
distribution_statement				X
document_list	X			
document_usage_parameter				X
subject				X
value_of				X
drawing				X
heading				X
pages				X
drawing_suffix_number_combination	X			
drawing_number	X			
suffix_identifier			X	
effectivity				X
description				X
effectivity_approvals				X
name				X
product				X
work_activity				X
element_identification	X			
alternate_identifications	X			
change_status	X			
design_activities	X			
element_classifications		X		
identifying_number	X			

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
outstanding_changes	X			
title		X		
element_type		X		
code_administrator		X		
element_code		X		
type_of_coding_scheme		X		
event				X
actual_date				X
assignment				X
description				X
id				X
name				X
planned_date				X
responsible_person_organization				X
external_library_reference				X
description				X
external_id				X
library_type				X
header	X			
document_abstract	X			
document_keywords	X			
header_configuration	X			
language	X			
sheet_count	X			
size_of_sheet	X			

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
header_configuration_with_element_-identification	X			
data_configuration	X			
identification	X			
identifier	X			
indentured_data_list	X			
body	X			
header	X			
indentured_data_list_body	X			
indentured_tabulation	X			
standarization_documents_list	X			
notes_list				X
revision_history				X
indentured_data_list_entry	X			
available_from				X
effective_on				X
entry	X			
entry_configuration				X
entry_content_property				X
entry_item_change_level				X
entry_notes				X
indenture_level	X			
retrofit				X
special_conditions			X	
superceded_entry				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
indentured_data_list_header	X			
common_header	X			
procurement_references				X
indentured_data_list_tabulation	X			
method_of_entry_tabulation	X			
tabulation_of_entries	X			
indentured_list_by_document	X			
top_indentured_tdp_elements	X			
indentured_list_by_part			X	
top_indentured_items			X	
indentured_list_by_part_with_document_-references_to_parts			X	
top_indentured			X	
independent_property				X
allowed_units				X
description				X
id				X
property_source				X
independent_property_relationship				X
description				X
related_property				X
relating_property				X
relation_type				X
independent_property_usage				X
assigned_property				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
assigned_to				X
item			X	
alternates				X
conditions_defined_through_-constrained_document				X
conditions_defined_through_simple_-reference				X
identification			X	
item_configuration			X	
item_contexts				X
item_weight				X
notes			X	
shape (N/A)				
shape_aspects (N/A)				
special_conditions			X	
item_identification			X	
alternate_identifications			X	
change_status			X	
classifications			X	
design_activity			X	
identifying_number			X	
item_certification				X
nomenclature_or_name				X
source_information			X	
item_parent_to_item_child_relationship			X	
child			X	

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
parent			X	
item_parent_to_tdp_element_child_-relationship			X	
child			X	
parent			X	
item_type				X
code_administrator				X
item_code				X
type_of_coding_scheme				X
label	X			
lot_effectivity				X
lot_identification				X
lot_size				X
lot_unit_of_measure				X
measure				X
notation	X			
note	X			
note_parameter				X
note_title	X			
reference_code	X			
referenced_document_in_note				X
type_of_notation	X			
number_with_units				X
units_of				X
value_of				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
other_reference_document				X
person				X
first_name				X
last_name				X
middle_name				X
title				X
unique_identifier				X
person_and_organization	X			
electronic_mail_address				X
organization	X			
person_address				X
person_identification				X
product_configuration		X		
approval_of_configuration		X		
assigned_to				X
configuration_id		X		
design_item		X		
product_name		X		
product_data_set				X
file_configuration				X
related				X
product_data_set_with_format				X
presentation_of_product_data_set				X
product_header				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
product_data_set_with_shading				X
a_shaded_model				X
data_configuration				X
product_data_set_without_format				X
data_configuration				X
a_model (N/A)				
product_model		X		
model_name		X		
projects				X
project				X
description				X
end_date				X
existence				X
id				X
name				X
participants				X
project_owner				X
start_date				X
property_assignment				X
described_element				X
description				X
name				X
quantity				X
quantity_accuracy				X
value_of				X

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
reference_document	X			
document_identification	X			
reference_document_usage	X			
document_referenced	X			
location_in_document	X			
usage_description	X			
usage_code	X			
usage_parameter	X			
relative_event				X
offset				X
related_event				X
release_authentication	X	X		X
authentication		X		
person_responsible				X
release_authority				X
release_authority_code				X
release_date	X			
retrofit_usage				X
defining_document				X
disposition_for				X
retrofit_description				X
revision	X			
revision_approval				X
revision_date	X			
revision_documents	X			

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
revision_description				X
revision_level	X			
security_classification				X
classification_date				X
classifier				X
declassification_date				X
item_classification				X
title_security_classification				X
sequence_effectivity				X
from_effectivity_id				X
quantity_unit_of_measure				X
thru_effectivity_id				X
total_component_quantity_in_product_configuration				X
sheet	X			
presentation_format (N/A)	X			
sheet_configuration	X			
sheet_size	X			
sheet_identifier	X			
size_characteristics		X		
byte_size		X		
sheet_size		X		
size_characteristics_full_size		X		
source_information_type		X		
source_code		X		

Table K.6.3 — Levels of capability for IDL application objects and attributes (continued)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
source_description		X		
type_of_coding_scheme		X		
source_location				X
path_information				X
storage_node_identification				X
special_condition		X		
code		X		
description		X		
type_of_coding_scheme		X		
specification_document				X
standard_document				X
status				X
status_code				X
status_code_basis				X
tdp_element	X			
tdp_element_and_item_association	X			
related_item	X			
related_tdp_element	X			
tdp_element_parent_to_item_child_-relationship			X	
child			X	
parent			X	
tdp_element_parent_to_tdp_element_child_-relationship	X			
child	X			
parent	X			

Table K.6.3 — Levels of capability for IDL application objects and attributes (concluded)

Indentured data list OBJECTS	Levels of capability			
	1	2	3	4
tdp_indentured_item			X	
top_indenture			X	
tdp_indentured_tdp_element	X			
top_indenture	X			
text	X			
time	X			
hour	X			
minute	X			
second	X			
zone	X			
time_interval_effectivity				**X**
duration				**X**
interval_name				**X**
primary_bound				**X**
secondary_bound				**X**
volume				**X**
weight				**X**

Bibliography

[1] ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML*)

[2] ISO 10744:1997, *Information technology — Hypermedia/Time-based Structuring Language (HyTime)*

[3] ISO 10179 :1996, *Information technology — Processing languages — Document Style Semantics and Specification Language (DSSSL)*

[4] ISO 10303-32:1998, *Industrial automation systems and integration — Product data representation and exchange - Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients*

[5] ISO 10303-209:2001, *Industrial automation systems and integration — Product data representation and exchange — Part 209: Application protocol: Composite and metallic structural analysis and related design*

[6] ISO/TS 10303-332:2002, *Industrial automation systems and integration — Product data representation and exchange — Part 332: Abstract test suite: Technical data packaging core information and exchange*

[7] ANSI X.12, *Electronic Data Interchange*; American National Standards Institute

[8] EDIFACT, *Electronic Data Interchange for Adminstration, Commerce, and Transportation*, (the series), 1995

[9] *Automated Interchange of Technical Information*, (MIL-STD-1840C), United States Military Standard

[10] *Functional Needs IDEF0 Activity and Information Models for the PAS-C Program*, 9 January 1992, Contract No. F33615-91-C-5713, Document No. PASC006-01.00, Air Force Systems Command, Department of the Air Force, Wright-Patterson AFB, Ohio 45433-6533

[11] *IDEF0 (ICAM Definition Language 0)*, Federal Information Processing Standards Publication 183, Integration Definition for Function Modeling (IDEF0), FIPS PUB 183, National Institute of Standards and Technology, December 1993

Index

A_number
application assertion 285, 286, 294
application object 55
ARM diagrams 1304
mapping table 312, 853
A_real
application assertion 247, 264, 268, 285, 287, 288, 308, 309
application object 56
ARM diagrams 1304
mapping table 312, 853
AAM
definition 14
access_file_requires_external_identification
AIM EXPRESS listing rules 1152
AIM EXPRESS short listing rules 979
Accessed_file
application assertion 239, 248, 249, 300
application object 56
ARM diagrams 1278
mapping table 389
action
AIM diagrams 1321
AIM EXPRESS listing entities 1061
action_assignment
AIM diagrams 1322
AIM EXPRESS listing entities 1061
action_directive
AIM diagrams 1363
AIM EXPRESS listing entities 1061
action_item
AIM EXPRESS listing types 1045
AIM EXPRESS short listing types 952
action_method
AIM diagrams 1321
AIM EXPRESS listing entities 1061
action_method_with_associated_documents
AIM diagrams 1321
AIM EXPRESS listing entities 1061
action_property
AIM diagrams 1322
AIM EXPRESS listing entities 1061
action_property_representation
AIM diagrams 1322
AIM EXPRESS listing entities 1061

action_relationship
AIM diagrams . . . 1321
AIM EXPRESS listing entities . . . 1061
action_request_assignment
AIM diagrams . . . 1363
AIM EXPRESS listing entities . . . 1062
action_request_item
AIM EXPRESS listing types . . . 1045
AIM EXPRESS short listing types . . . 952
action_request_solution
AIM diagrams . . . 1363
AIM EXPRESS listing entities . . . 1062
action_request_status
AIM diagrams . . . 1363
AIM EXPRESS listing entities . . . 1062
action_resource
AIM diagrams . . . 1364
AIM EXPRESS listing entities . . . 1062
action_resource_relationship
AIM diagrams . . . 1364
AIM EXPRESS listing entities . . . 1062
action_resource_type
AIM diagrams . . . 1364
AIM EXPRESS listing entities . . . 1062
action_status
AIM diagrams . . . 1321
AIM EXPRESS listing entities . . . 1062
mapping table . . . 384
acyclic_composite_text
AIM EXPRESS listing functions . . . 1169
acyclic_curve_replica
AIM EXPRESS listing functions . . . 1170
acyclic_mapped_representation
AIM EXPRESS listing functions . . . 1170
acyclic_point_replica
AIM EXPRESS listing functions . . . 1171
acyclic_product_category_relationship
AIM EXPRESS listing functions . . . 1171
acyclic_product_definition_relationship
AIM EXPRESS listing functions . . . 1172
acyclic_solid_replica
AIM EXPRESS listing functions . . . 1172
acyclic_surface_replica
AIM EXPRESS listing functions . . . 1172
address
AIM diagrams . . . 1333
AIM EXPRESS listing entities . . . 1062

advanced_brep_shape_representation
AIM diagrams . . . 1315
AIM EXPRESS listing entities . . . 1063
mapping table . . . 586, 919
advanced_face
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1064
mapping table . . . 919
ahead_or_behind
AIM EXPRESS listing types . . . 1045
AIC
definition . . . 14
AIM
definition . . . 14
Alternate_element_identification
application assertion . . . 239, 240, 262
application object . . . 57
ARM diagrams . . . 1292
mapping table . . . 312
Alternate_identification_of_item
application assertion . . . 240, 241, 280
application object . . . 59
ARM diagrams . . . 1292
mapping table . . . 389, 519, 566, 594, 749, 874
Alternate_item
application assertion . . . 241, 278
application object . . . 60
ARM diagrams . . . 1288
mapping table . . . 394, 598, 753
alternate_product_relationship
AIM diagrams . . . 1308
AIM EXPRESS listing entities . . . 1065
amount_of_substance_measure
AIM EXPRESS listing types . . . 1045
amount_of_substance_measure_with_unit
AIM diagrams . . . 1340
AIM EXPRESS listing entities . . . 1065
amount_of_substance_unit
AIM diagrams . . . 1338
AIM EXPRESS listing entities . . . 1065
An_integer
application assertion . . . 241, 259, 269, 300, 301, 308
application object . . . 62
ARM diagrams . . . 1304
mapping table . . . 317, 853

Annotation
application assertion 241, 293
application object 62
ARM diagrams 1301
mapping table 570, 878
annotation_curve_occurrence
AIM diagrams 1319
AIM EXPRESS listing entities 1066
annotation_occurrence
AIM diagrams 1319
AIM EXPRESS listing entities 1066
mapping table 570, 855, 878
annotation_point_occurrence
AIM diagrams 1319
AIM EXPRESS listing entities 1066
annotation_text
AIM diagrams 1318
AIM EXPRESS listing entities 1066
annotation_text_character
AIM diagrams 1318
AIM EXPRESS listing entities 1066
annotation_text_occurrence
AIM diagrams 1319
AIM EXPRESS listing entities 1066
mapping table 856-858, 872
AP
definition 14
application_context
AIM diagrams 1307
AIM EXPRESS listing entities 1067
application_context_element
AIM diagrams 1307
AIM EXPRESS listing entities 1067
application_context_relationship
AIM diagrams 1307
AIM EXPRESS listing entities 1067
application_protocol_definition
AIM diagrams 1307
AIM EXPRESS listing entities 1067
applied_action_assignment
AIM diagrams 1322
AIM EXPRESS listing entities 1067, 1068
AIM EXPRESS short listing entities 960
mapping table 672, 830
applied_action_request_assignment
AIM diagrams 1364
AIM EXPRESS listing entities 1067
AIM EXPRESS short listing entities 961

applied_approval_assignment
AIM diagrams 1324
AIM EXPRESS listing entities 1067
AIM EXPRESS short listing entities 961
mapping table 317
applied_certification_assignment
AIM diagrams 1370
AIM EXPRESS listing entities 1067
AIM EXPRESS short listing entities 961
applied_classification_assignment
AIM diagrams 1344
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 962
applied_contract_assignment
AIM diagrams 1339
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 962
mapping table 400, 525, 605
applied_date_and_time_assignment
AIM diagrams 1336
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 963
applied_date_assignment
AIM diagrams 1337
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 963
applied_document_reference
AIM diagrams 1327
AIM EXPRESS short listing entities 964
mapping table 385, 443, 469, 470, 472, 482, 503, 640, 643-645, 654
applied_document_usage_constraint_assignment
AIM diagrams 1326
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 964
applied_document_usage_constraint_assignment_restriction
AIM EXPRESS listing rules 1153
AIM EXPRESS short listing rules 980
applied_effectivity_assignment
AIM diagrams 1329
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 965
mapping table 450, 610, 762
applied_effectivity_context_assignment
AIM diagrams 1329
AIM EXPRESS listing entities 1068
AIM EXPRESS short listing entities 965
mapping table 450, 610, 762

applied_event_occurrence_assignment
AIM diagrams . . . 1365
AIM EXPRESS listing entities . . . 1068
AIM EXPRESS short listing entities . . . 966
applied_external_identification_assignment
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1068
AIM EXPRESS short listing entities . . . 966
mapping table . . . 506, 562, 683
applied_identification_assignment
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1068
AIM EXPRESS short listing entities . . . 967
applied_organization_assignment
AIM diagrams . . . 1334
AIM EXPRESS listing entities . . . 1068
AIM EXPRESS short listing entities . . . 967
applied_organizational_project_assignment
AIM diagrams . . . 1334, 1335
AIM EXPRESS listing entities . . . 1069
AIM EXPRESS short listing entities . . . 967
applied_person_and_organization_assignment
AIM diagrams . . . 1332
AIM EXPRESS listing entities . . . 1069
AIM EXPRESS short listing entities . . . 968
applied_presented_item
AIM diagrams . . . 1315
AIM EXPRESS listing entities . . . 1069
AIM EXPRESS short listing entities . . . 968
applied_security_classification_assignment
AIM diagrams . . . 1331
AIM EXPRESS listing entities . . . 1069
AIM EXPRESS short listing entities . . . 969
approval
AIM diagrams . . . 1324
AIM EXPRESS listing entities . . . 1069
mapping table . . . 346, 378
Approval
application assertion . . . 241-243, 247, 261, 283, 291
application object . . . 62
ARM diagrams . . . 1295
mapping table . . . 317
approval_assignment
AIM diagrams . . . 1324
AIM EXPRESS listing entities . . . 1069
approval_date_time
AIM diagrams . . . 1324
AIM EXPRESS listing entities . . . 1069

approval_item
AIM EXPRESS listing types 1045
AIM EXPRESS short listing types 953
approval_person_organization
AIM diagrams 1324
AIM EXPRESS listing entities 1069
approval_relationship
AIM diagrams 1324
AIM EXPRESS listing entities 1069
approval_requires_approval_date_time
AIM EXPRESS listing rules 1153
AIM EXPRESS short listing rules 981
approval_requires_approval_person_organization
AIM EXPRESS listing rules 1153
AIM EXPRESS short listing rules 982
approval_role
AIM diagrams 1324
AIM EXPRESS listing entities 1070
approval_status
AIM diagrams 1324
AIM EXPRESS listing entities 1070
approved_item
AIM EXPRESS listing types 1046
area_dependent_annotation_representation
AIM diagrams 1320
AIM EXPRESS listing entities 1070
mapping table 853, 863-865, 872
area_in_set
AIM diagrams 1320
AIM EXPRESS listing entities 1070
area_measure
AIM EXPRESS listing types 1046
area_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1070
area_or_view
AIM EXPRESS listing types 1046
area_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1070
ARM
definition 14
aspect_ratio
AIM EXPRESS listing functions 1173
assembly_component_usage
AIM diagrams 1310
AIM EXPRESS listing entities 1070
mapping table 757, 812

assembly_component_usage_substitute
AIM diagrams . . . 1310
AIM EXPRESS listing entities . . . 1071
assembly_component_usage_substitute_with_ranking
AIM diagrams . . . 1310
AIM EXPRESS listing entities . . . 1071
Assembly_relationship
application assertion . . . 302
application object . . . 63
ARM diagrams . . . 1282
mapping table . . . 757
associated list
definition . . . 10
Associated_list
application assertion . . . 242
application object . . . 64
ARM diagrams . . . 1273
mapping table . . . 523, 602, 694, 719, 757
associated_surface
AIM EXPRESS listing functions . . . 1173
axis1_placement
AIM diagrams . . . 1345
AIM EXPRESS listing entities . . . 1071
axis2_placement
AIM EXPRESS listing types . . . 1046
mapping table . . . 803, 849
axis2_placement_2d
AIM diagrams . . . 1345
AIM EXPRESS listing entities . . . 1071
mapping table . . . 858, 871
axis2_placement_3d
AIM diagrams . . . 1345
AIM EXPRESS listing entities . . . 1071
b_spline_curve
AIM diagrams . . . 1349
AIM EXPRESS listing entities . . . 1071
b_spline_curve_form
AIM EXPRESS listing types . . . 1046
b_spline_curve_with_knots
AIM diagrams . . . 1349
AIM EXPRESS listing entities . . . 1072
b_spline_surface
AIM diagrams . . . 1353
AIM EXPRESS listing entities . . . 1072
b_spline_surface_form
AIM EXPRESS listing types . . . 1046
b_spline_surface_with_knots
AIM diagrams . . . 1353
AIM EXPRESS listing entities . . . 1073

background_colour
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1073
bag_to_set
AIM EXPRESS listing functions . . . 1173
base_axis
AIM EXPRESS listing functions . . . 1173
bezier_curve
AIM diagrams . . . 1349
AIM EXPRESS listing entities . . . 1073
bezier_surface
AIM diagrams . . . 1353
AIM EXPRESS listing entities . . . 1073
block
AIM diagrams . . . 1357
AIM EXPRESS listing entities . . . 1073
Body_presentation
application assertion . . . 242, 284
application object . . . 65
ARM diagrams . . . 1299
mapping table . . . 853
BOM
definition . . . 14
boolean_choose
AIM EXPRESS listing functions . . . 1174
boolean_operand
AIM EXPRESS listing types . . . 1046
boolean_operator
AIM EXPRESS listing types . . . 1046
boolean_result
AIM diagrams . . . 1357
AIM EXPRESS listing entities . . . 1073
boundary_curve
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1073
bounded_curve
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1074
bounded_pcurve
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1074
bounded_surface
AIM diagrams . . . 1352
AIM EXPRESS listing entities . . . 1074
bounded_surface_curve
AIM diagrams . . . 1350
AIM EXPRESS listing entities . . . 1074

box_domain
AIM diagrams 1357
AIM EXPRESS listing entities 1074
boxed_half_space
AIM diagrams 1357
AIM EXPRESS listing entities 1074
brep_with_voids
AIM diagrams 1355
AIM EXPRESS listing entities 1074
build_2axes
AIM EXPRESS listing functions 1174
build_axes
AIM EXPRESS listing functions 1174
CAD
definition 14
calendar_date
AIM diagrams 1336
AIM EXPRESS listing entities 1074
camera_image
AIM diagrams 1314
AIM EXPRESS listing entities 1074
camera_image_3d_with_scale
AIM diagrams 1314
AIM EXPRESS listing entities 1075
camera_model
AIM diagrams 1316
AIM EXPRESS listing entities 1075
camera_model_d3
AIM diagrams 1317
AIM EXPRESS listing entities 1075
camera_model_d3_with_hlhsr
AIM diagrams 1317
AIM EXPRESS listing entities 1076
camera_model_with_light_sources
AIM diagrams 1317
AIM EXPRESS listing entities 1076
camera_usage
AIM diagrams 1314
AIM EXPRESS listing entities 1076
cartesian_point
AIM diagrams 1346
AIM EXPRESS listing entities 1076
mapping table 868
cartesian_transformation_operator
AIM diagrams 1345
AIM EXPRESS listing entities 1076
cartesian_transformation_operator_2d
AIM diagrams 1367
AIM EXPRESS listing entities 1076

cartesian_transformation_operator_3d
AIM diagrams 1345
AIM EXPRESS listing entities 1076
CDRL
definition 14
celsius_temperature_measure
AIM EXPRESS listing types 1047
central_or_parallel
AIM EXPRESS listing types 1047
Certification
application assertion 242, 243, 262, 280
application object 65
ARM diagrams 1295
mapping table 319
certification
AIM diagrams 1370
AIM EXPRESS listing entities 1077
mapping table 319
certification_assignment
AIM diagrams 1370
AIM EXPRESS listing entities 1077
certification_item
AIM EXPRESS listing types 1047
certification_type
AIM diagrams 1370
AIM EXPRESS listing entities 1077
Change_identification
application assertion 239, 240, 243-245, 262, 266, 280
application object 66
ARM diagrams 1291
mapping table 320
change_identification_restricts_executed_action
AIM EXPRESS listing rules 1153
AIM EXPRESS short listing rules 983
character_glyph_style_outline
AIM diagrams 1319
AIM EXPRESS listing entities 1077
character_glyph_style_stroke
AIM diagrams 1319
AIM EXPRESS listing entities 1077
character_glyph_symbol
AIM diagrams 1315
AIM EXPRESS listing entities 1077
character_style_select
AIM EXPRESS listing types 1047
characterized_action_definition
AIM EXPRESS listing types 1047
characterized_definition
AIM EXPRESS listing types 1047

characterized_object
AIM diagrams . . . 1311
AIM EXPRESS listing entities . . . 1077
characterized_product_definition
AIM EXPRESS listing types . . . 1047
CID
definition . . . 14
circle
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1077
class
AIM diagrams . . . 1323
AIM EXPRESS listing entities . . . 1078
AIM EXPRESS short listing entities . . . 969
class_system
AIM diagrams . . . 1323
AIM EXPRESS listing entities . . . 1078
AIM EXPRESS short listing entities . . . 969
classification_assignment
AIM diagrams . . . 1344
AIM EXPRESS listing entities . . . 1078
classification_item
AIM EXPRESS listing types . . . 1047
AIM EXPRESS short listing types . . . 953
classification_role
AIM diagrams . . . 1344
AIM EXPRESS listing entities . . . 1078
closed_shell
AIM diagrams . . . 1355
AIM EXPRESS listing entities . . . 1078
closed_shell_reversed
AIM EXPRESS listing functions . . . 1175
CM
definition . . . 14
colour
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1078
colour_rgb
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1078
colour_specification
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1078
Column_header
application assertion . . . 245, 306
application object . . . 72
ARM diagrams . . . 1299
mapping table . . . 853

Columnar_data_content_holder
application assertion . . . 245, 306
application object . . . 72
ARM diagrams . . . 1299
mapping table . . . 853
commercial drawings
definition . . . 10
Company
application assertion . . . 245, 259, 263, 282, 291, 296, 305
application object . . . 73
ARM diagrams . . . 1297
mapping table . . . 331
Company_code
application assertion . . . 245, 259, 296, 306
application object . . . 73
ARM diagrams . . . 1293
mapping table . . . 332
compatible_dimension
AIM EXPRESS listing rules . . . 1154
AIM EXPRESS short listing rules . . . 984
composite_curve
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1078
composite_curve_on_surface
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1079
composite_curve_segment
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1079
composite_text
AIM diagrams . . . 1318
AIM EXPRESS listing entities . . . 1079
compound_item_definition
AIM EXPRESS listing types . . . 1047
compound_representation_item
AIM diagrams . . . 1314
AIM EXPRESS listing entities . . . 1079
computer interpretable data
definition . . . 10
computer processable data
definition . . . 10
conceptual design drawings
definition . . . 10
conditional_reverse
AIM EXPRESS listing functions . . . 1175

Configuration
application assertion 246, 270, 271, 278, 299
application object 74
ARM diagrams 1290
mapping table 335
configuration_design
AIM diagrams 1328
AIM EXPRESS listing entities 1079
configuration_design_item
AIM EXPRESS listing types 1047
configuration_effectivity
AIM diagrams 1328
AIM EXPRESS listing entities 1079
mapping table 450, 610, 762
configuration_item
AIM diagrams 1331
AIM EXPRESS listing entities 1080
mapping table 374
conic
AIM diagrams 1347
AIM EXPRESS listing entities 1080
conical_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1080
connected_edge_set
AIM diagrams 1354
AIM EXPRESS listing entities 1080
connected_face_set
AIM diagrams 1355
AIM EXPRESS listing entities 1080
constraints_composite_curve_on_surface
AIM EXPRESS listing functions 1175
constraints_geometry_shell_based_surface_model
AIM EXPRESS listing functions 1176
constraints_geometry_shell_based_wireframe_model
AIM EXPRESS listing functions 1176
constraints_param_b_spline
AIM EXPRESS listing functions 1176
constraints_rectangular_composite_surface
AIM EXPRESS listing functions 1177
Content_property
application assertion 247, 249, 266, 272
application object 76
ARM diagrams 1296
mapping table 604, 879
context_dependent_measure
AIM EXPRESS listing types 1047

context_dependent_shape_representation
AIM diagrams . . . 1367
AIM EXPRESS listing entities . . . 1080
context_dependent_unit
AIM diagrams . . . 1338
AIM EXPRESS listing entities . . . 1080
Contract
application assertion . . . 243, 246, 247
application object . . . 77
ARM diagrams . . . 1290
mapping table . . . 340
contract
AIM diagrams . . . 1339
AIM EXPRESS listing entities . . . 1080
mapping table . . . 340
contract_assignment
AIM diagrams . . . 1339
AIM EXPRESS listing entities . . . 1081
contract_item
AIM EXPRESS listing types . . . 1047
AIM EXPRESS short listing types . . . 954
contract_relationship
AIM diagrams . . . 1339
AIM EXPRESS listing entities . . . 1081
Contract_submission
application assertion . . . 247, 248, 250, 257, 272
application object . . . 78
ARM diagrams . . . 1279
mapping table . . . 400, 525, 605
contract_submission_requires_date_and_organization
AIM EXPRESS short listing rules . . . 985
contract_type
AIM diagrams . . . 1339
AIM EXPRESS listing entities . . . 1081
conversion_based_unit
AIM diagrams . . . 1338
AIM EXPRESS listing entities . . . 1081
coordinated_universal_time_offset
AIM diagrams . . . 1337
AIM EXPRESS listing entities . . . 1081
count_measure
AIM EXPRESS listing types . . . 1048
cross_product
AIM EXPRESS listing functions . . . 1178
csg_primitive
AIM EXPRESS listing types . . . 1048
csg_select
AIM EXPRESS listing types . . . 1048

csg_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1081
mapping table 586, 919
csg_solid
AIM diagrams 1357
AIM EXPRESS listing entities 1082
curve
AIM diagrams 1347
AIM EXPRESS listing entities 1082
curve_bounded_surface
AIM diagrams 1352
AIM EXPRESS listing entities 1082
curve_font_or_scaled_curve_font_select
AIM EXPRESS listing types 1048
curve_on_surface
AIM EXPRESS listing types 1048
curve_or_render
AIM EXPRESS listing types 1048
curve_replica
AIM diagrams 1347
AIM EXPRESS listing entities 1082
curve_style
AIM diagrams 1361
AIM EXPRESS listing entities 1083
curve_style_font
AIM diagrams 1361
AIM EXPRESS listing entities 1083
curve_style_font_pattern
AIM diagrams 1361
AIM EXPRESS listing entities 1083
curve_style_font_select
AIM EXPRESS listing types 1048
curve_style_rendering
AIM diagrams 1361
AIM EXPRESS listing entities 1083
curve_weights_positive
AIM EXPRESS listing functions 1178
cylindrical_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1083
DAG
definition 14
data definition exchange
definition 11
data list
definition 11

Data_definition_entry_item
application assertion 248, 249, 254
application object 79
ARM diagrams 1275
mapping table 402
data_definition_entry_string_restrict_for_superseded_element
AIM EXPRESS listing rules 1154
AIM EXPRESS short listing rules 988
Data_definition_entry_tdp_element
application assertion 249-251, 254
application object 82
ARM diagrams 1276
mapping table 413
Data_definition_exchange
application assertion 252
application object 88
ARM diagrams 1273
mapping table 429
Unit of Functionality 26
Data_definition_exchange_body
application assertion 252, 253
application object 89
ARM diagrams 1275
mapping table 430
data_definition_exchange_categorized_as_document
AIM EXPRESS listing rules 1155
AIM EXPRESS short listing rules 989
Data_definition_exchange_header
application assertion 252-254
application object 90
ARM diagrams 1274
mapping table 435
data_definition_exchange_restrict_to_one_presentation
AIM EXPRESS listing rules 1156
AIM EXPRESS short listing rules 990
Data_definition_exchange_simple_entry
application assertion 254, 300
application object 92
ARM diagrams 1275
mapping table 439
Data_definition_indentured_entry
application assertion 254-256
application object 92
ARM diagrams 1275
mapping table 440

Data_definition_indentured_list_method
application assertion 252, 256
application object 95
ARM diagrams 1275
mapping table 443
Data_list
application assertion 256, 257
application object 96
ARM diagrams 1283
mapping table 527, 696
Unit of Functionality 31
Data_list_body
application assertion 256, 257
application object 96
ARM diagrams 1283
mapping table 528
Data_list_entry
application assertion 257, 258
application object 97
ARM diagrams 1283
mapping table 530
Data_list_header
application assertion 257, 258
application object 99
ARM diagrams 1283
mapping table 536, 696
Data_list_tabulation
application assertion 257, 258
application object 99
ARM diagrams 1283
mapping table 536
Date
application assertion 241-243, 247, 248, 253, 259, 263, 267, 293, 297, 298, 308
application object 100
ARM diagrams 1305
mapping table 344, 854
date
AIM diagrams 1336
AIM EXPRESS listing entities 1083
mapping table 344, 345, 854
date_and_time
AIM diagrams 1336
AIM EXPRESS listing entities 1083
mapping table 344, 854
date_and_time_assignment
AIM diagrams 1336
AIM EXPRESS listing entities 1083

date_and_time_item
AIM EXPRESS listing types 1048
AIM EXPRESS short listing types 954
date_assignment
AIM diagrams 1337
AIM EXPRESS listing entities 1083
Date_effectivity
application assertion 259
application object 101
ARM diagrams 1294
mapping table 447, 607, 759
date_item
AIM EXPRESS listing types 1049
AIM EXPRESS short listing types 955
date_role
AIM diagrams 1337
AIM EXPRESS listing entities 1083
date_time_or_event_occurrence
AIM EXPRESS listing types 1049
date_time_role
AIM diagrams 1336
AIM EXPRESS listing entities 1084
date_time_select
AIM EXPRESS listing types 1049
dated_effectivity
AIM diagrams 1328
AIM EXPRESS listing entities 1084
mapping table 447, 450, 607, 610, 759, 762
day_in_month_number
AIM EXPRESS listing types 1049
DDE
definition 14
defined_character_glyph
AIM diagrams 1318
AIM EXPRESS listing entities 1084
defined_glyph_select
AIM EXPRESS listing types 1049
defined_symbol_select
AIM EXPRESS listing types 1049
definitional_representation
AIM diagrams 1313
AIM EXPRESS listing entities 1084
degenerate_pcurve
AIM diagrams 1346
AIM EXPRESS listing entities 1084
degenerate_toroidal_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1084

Delivery_accounting
application assertion 248, 250, 253, 259
application object 102
ARM diagrams 1274
mapping table 448
dependent_instantiable_named_unit
AIM EXPRESS listing rules 1156
AIM EXPRESS short listing rules 991
derive_dimensional_exponents
AIM EXPRESS listing functions 1178
derived_property_select
AIM EXPRESS listing types 1049
derived_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1084
derived_unit_element
AIM diagrams 1338
AIM EXPRESS listing entities 1085
description_attribute
AIM diagrams 1342
AIM EXPRESS listing entities 1085
description_attribute_select
AIM EXPRESS listing types 1049
descriptive_measure
AIM EXPRESS listing types 1050
descriptive_representation_item
AIM diagrams 1314
AIM EXPRESS listing entities 1085
mapping table 485, 506, 546, 562, 657, 684, 709, 727, 809, 840
design activity standard
definition 11
design disclosure
definition 11
Design_authority
application assertion 239, 240, 259, 260, 262, 280
application object 103
ARM diagrams 1289
mapping table 345
design_make_from_relationship
AIM diagrams 1310
AIM EXPRESS listing entities 1085
AIM EXPRESS short listing entities 970
mapping table 780
developmental design drawings
definition 11
DID
definition 14
dimension_count
AIM EXPRESS listing types 1050

dimension_of
AIM EXPRESS listing functions 1179
dimensional_exponents
AIM diagrams 1338
AIM EXPRESS listing entities 1085
dimensions_for_si_unit
AIM EXPRESS listing functions 1180
directed_action
AIM diagrams 1363
AIM EXPRESS listing entities 1085
direction
AIM diagrams 1345
AIM EXPRESS listing entities 1085
direction_count_select
AIM EXPRESS listing types 1050
Distribution_notice
application assertion 246, 260, 266
application object 104
ARM diagrams 1290
mapping table 346
distribution_notice_approval_requires_supporting_data
AIM EXPRESS listing rules 1157
AIM EXPRESS short listing rules 992
DL
definition 14
document
AIM diagrams 1325
AIM EXPRESS listing entities 1085
mapping table 607
document_file
AIM EXPRESS listing entities 1085
AIM EXPRESS short listing entities 970
mapping table 389, 457, 459, 770, 860, 884, 926
Document_list
application object 105
ARM diagrams 1273
mapping table 537, 607, 697
document_product_association
AIM diagrams 1327
AIM EXPRESS listing entities 1086
Document_product_equivalence
AIM diagrams 1327
AIM EXPRESS listing entities 1086
AIM EXPRESS short listing entities 971
document_product_equivalence_existence_rule
AIM EXPRESS short listing rules 993
document_reference
AIM diagrams 1327
AIM EXPRESS listing entities 1086

document_reference_item
AIM EXPRESS listing types 1050
AIM EXPRESS short listing types 955
document_relationship
AIM diagrams 1325
AIM EXPRESS listing entities 1086
mapping table 468, 779, 863, 892
document_representation_type
AIM diagrams 1325
AIM EXPRESS listing entities 1087
document_type
AIM diagrams 1325
AIM EXPRESS listing entities 1087
document_usage_constraint
AIM diagrams 1325
AIM EXPRESS listing entities 1087
mapping table 347, 376, 448
document_usage_constraint_assignment
AIM diagrams 1325
AIM EXPRESS listing entities 1087
Document_usage_parameter
application assertion 260, 296
application object 105
ARM diagrams 1303
mapping table 347, 448
document_usage_role
AIM diagrams 1325
AIM EXPRESS listing entities 1087
dot_product
AIM EXPRESS listing functions 1180
draughting_approval_assignment
AIM diagrams 1324
AIM EXPRESS listing entities 1087
draughting_callout
AIM diagrams 1344
AIM EXPRESS listing entities 1087
draughting_callout_element
AIM EXPRESS listing types 1050
draughting_drawing_revision
AIM diagrams 1320
AIM EXPRESS listing entities 1087
mapping table 530, 538, 565, 571, 608, 686, 721, 747, 760, 846, 881
draughting_pre_defined_colour
AIM diagrams 1360
AIM EXPRESS listing entities 1090
draughting_pre_defined_curve_font
AIM diagrams 1360
AIM EXPRESS listing entities 1090

draughting_title
AIM diagrams 1320
AIM EXPRESS listing entities 1090
draughting_titled_item
AIM EXPRESS listing types 1050
Drawing
application assertion 242, 261, 292
application object 106
ARM diagrams 1301
mapping table 538, 571, 608, 697, 721, 760, 881
Unit of Functionality 34
drawing types
definition 11
drawing_definition
AIM diagrams 1320
AIM EXPRESS listing entities 1090
drawing_revision
AIM diagrams 1320
AIM EXPRESS listing entities 1090
drawing_sheet_revision
AIM diagrams 1320
AIM EXPRESS listing entities 1090
mapping table 501, 530, 560, 590, 681, 744, 838
Drawing_suffix_number_combination
application assertion 240, 261, 280
application object 106
ARM diagrams 1287
mapping table 448, 539, 576, 609, 761, 882
drawing_suffix_number_combination_identification_constraint
AIM EXPRESS listing rules 1157
AIM EXPRESS short listing rules 995
DWG
definition 14
edge
AIM diagrams 1354
AIM EXPRESS listing entities 1090
edge_based_wireframe_model
AIM diagrams 1356
AIM EXPRESS listing entities 1091
edge_based_wireframe_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1091
mapping table 586, 919
edge_curve
AIM diagrams 1354
AIM EXPRESS listing entities 1092
edge_loop
AIM diagrams 1354
AIM EXPRESS listing entities 1092

edge_reversed
AIM EXPRESS listing functions 1181
Effectivity
application assertion 254, 261, 272, 283
application object 107
ARM diagrams 1294
mapping table 450, 610, 762
effectivity
AIM diagrams 1328
AIM EXPRESS listing entities 1092
effectivity_assignment
AIM diagrams 1329
AIM EXPRESS listing entities 1092
effectivity_context_assignment
AIM diagrams 1329
AIM EXPRESS listing entities 1092
effectivity_context_item
AIM EXPRESS listing types 1051
AIM EXPRESS short listing types 956
effectivity_context_role
AIM diagrams 1329
AIM EXPRESS listing entities 1092
effectivity_item
AIM EXPRESS listing types 1051
AIM EXPRESS short listing types 956
effectivity_relationship
AIM diagrams 1328
AIM EXPRESS listing entities 1093
electric_current_measure
AIM EXPRESS listing types 1051
electric_current_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1093
electric_current_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1093
Element_identification
application assertion 250, 253, 254, 261-263, 270-272, 274, 282, 287, 295, 299, 307, 308
application object 108
ARM diagrams 1289
mapping table 348
Element_type
application assertion 239, 262, 263
application object 111
ARM diagrams 1293
mapping table 354

elementary_brep_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1093
mapping table 586, 919
elementary_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1095
ellipse
AIM diagrams 1347
AIM EXPRESS listing entities 1095
evaluated_degenerate_pcurve
AIM diagrams 1346
AIM EXPRESS listing entities 1095
Event
application assertion 263, 264, 296, 309
application object 111
ARM diagrams 1297
mapping table 453, 614, 766
event_occurrence
AIM diagrams 1365
AIM EXPRESS listing entities 1095
mapping table 453, 455, 456, 614-616, 766, 768, 769
event_occurrence_assignment
AIM diagrams 1365
AIM EXPRESS listing entities 1095
event_occurrence_item
AIM EXPRESS listing types 1051
AIM EXPRESS short listing types 956
event_occurrence_relationship
AIM diagrams 1365
AIM EXPRESS listing entities 1095
event_occurrence_role
AIM diagrams 1365
AIM EXPRESS listing entities 1096
Exchange_file
application assertion 248, 250, 263, 264, 300
application object 113
ARM diagrams 1278
mapping table 457
exchange_reason_existence_required
AIM EXPRESS listing rules 1158
AIM EXPRESS short listing rules 996
executed_action
AIM diagrams 1321
AIM EXPRESS listing entities 1096
mapping table 320, 495, 555, 675, 712, 739, 832
existence_dependent_drawing_sheet_revision
AIM EXPRESS listing rules 1159
AIM EXPRESS short listing rules 998

Explicit_graphics
application assertion . . . 264, 289
application object . . . 115
ARM diagrams . . . 1298
mapping table . . . 855
External_graphics_file
application assertion . . . 264, 289
application object . . . 115
ARM diagrams . . . 1298
mapping table . . . 855
external_identification_assignment
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1096
external_identification_item
AIM EXPRESS listing types . . . 1051
AIM EXPRESS short listing types . . . 957
External_library_reference
application assertion . . . 264, 265, 276
application object . . . 116
ARM diagrams . . . 1302
mapping table . . . 617, 765, 883
external_source
AIM diagrams . . . 1330
AIM EXPRESS listing entities . . . 1096
externally_defined_character_glyph
AIM diagrams . . . 1318
AIM EXPRESS listing entities . . . 1096
externally_defined_class
AIM diagrams . . . 1323
AIM EXPRESS listing entities . . . 1096
AIM EXPRESS short listing entities . . . 972
mapping table . . . 458, 617, 765, 883
externally_defined_general_property
AIM diagrams . . . 1369
AIM EXPRESS listing entities . . . 1096
AIM EXPRESS short listing entities . . . 972
mapping table . . . 458, 617, 765, 883
externally_defined_item
AIM diagrams . . . 1330
AIM EXPRESS listing entities . . . 1096
externally_defined_planar_box
AIM diagrams . . . 1317
AIM EXPRESS listing entities . . . 1096
AIM EXPRESS short listing entities . . . 973
externally_defined_style
AIM diagrams . . . 1319
AIM EXPRESS listing entities . . . 1096

externally_defined_symbol
AIM diagrams . . . 1330
AIM EXPRESS listing entities . . . 1096
Externally_defined_symbol_and_placement
AIM diagrams . . . 1330
AIM EXPRESS listing entities . . . 1096
AIM EXPRESS short listing entities . . . 973
mapping table . . . 855
externally_defined_text_font
AIM diagrams . . . 1330
AIM EXPRESS listing entities . . . 1097
extruded_face_solid
AIM diagrams . . . 1355
AIM EXPRESS listing entities . . . 1097
face
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1097
face_based_surface_model
AIM diagrams . . . 1356
AIM EXPRESS listing entities . . . 1097
face_bound
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1097
face_bound_reversed
AIM EXPRESS listing functions . . . 1181
face_outer_bound
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1097
face_reversed
AIM EXPRESS listing functions . . . 1182
face_surface
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1097
faceted_brep
AIM diagrams . . . 1355
AIM EXPRESS listing entities . . . 1097
faceted_brep_shape_representation
AIM diagrams . . . 1315
AIM EXPRESS listing entities . . . 1097
mapping table . . . 586, 919
Field_content_holder
application assertion . . . 265
application object . . . 117
ARM diagrams . . . 1300
mapping table . . . 856

Field_content_holder_singular
application assertion 265, 270
application object 118
ARM diagrams 1300
mapping table 857
Field_content_holder_tabulation
application assertion 245, 265, 269
application object 118
ARM diagrams 1300
mapping table 857
Field_location
application assertion 265
application object 119
ARM diagrams 1300
mapping table 858
Field_location_definition
application assertion 265
application object 119
ARM diagrams 1300
mapping table 858
Field_title
application assertion 245, 265, 266, 269, 270, 307
application object 120
ARM diagrams 1300
mapping table 858
File
application assertion 264, 266-268, 279, 285
application object 121
ARM diagrams 1278
mapping table 459, 770, 860, 884
File_format
application assertion 250, 266-268
application object 123
ARM diagrams 1278
mapping table 466, 776, 862, 889
file_format_restricted_to_one_data_definition_entry
AIM EXPRESS listing rules 1159
AIM EXPRESS short listing rules 999
File_relationship
application assertion 268
application object 125
ARM diagrams 1279
mapping table 468, 779, 863, 892
fill_area_style
AIM diagrams 1360
AIM EXPRESS listing entities 1098
fill_area_style_colour
AIM diagrams 1360
AIM EXPRESS listing entities 1098

fill_style_select
AIM EXPRESS listing types 1051
first_proj_axis
AIM EXPRESS listing functions 1182
font_select
AIM EXPRESS listing types 1051
Foreign_defined_item
application object 126
ARM diagrams 1281
mapping table 780
founded_item
AIM diagrams 1317
AIM EXPRESS listing entities 1098
founded_item_select
AIM EXPRESS listing types 1051
functionally_defined_transformation
AIM diagrams 1312
AIM EXPRESS listing entities 1099
gbsf_check_curve
AIM EXPRESS listing functions 1183
gbsf_check_point
AIM EXPRESS listing functions 1184
gbsf_check_surface
AIM EXPRESS listing functions 1184
General_item_definition_relationship
application assertion 268
application object 126
ARM diagrams 1282
mapping table 780
general_property
AIM diagrams 1369
AIM EXPRESS listing entities 1099
mapping table 474, 475, 647, 648, 781, 782, 893, 894
general_property_association
AIM diagrams 1369
AIM EXPRESS listing entities 1099
mapping table 476, 649, 783, 895
general_property_relationship
AIM diagrams 1369
AIM EXPRESS listing entities 1099
mapping table 475, 476, 648, 649, 782, 783, 894, 895
geometric_curve_set
AIM diagrams 1346
AIM EXPRESS listing entities 1099
geometric_representation_context
AIM diagrams 1313
AIM EXPRESS listing entities 1099

geometric_representation_item
AIM diagrams 1316
AIM EXPRESS listing entities 1099
mapping table 576, 863, 893
geometric_set
AIM diagrams 1346
AIM EXPRESS listing entities 1100
geometric_set_select
AIM EXPRESS listing types 1051
Geometric_validation_property
application assertion 268, 269, 299
application object 127
ARM diagrams 1302
mapping table 892
geometrically_bounded_2d_wireframe_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1100
mapping table 586, 919
geometrically_bounded_surface_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1101
mapping table 586, 919
geometrically_bounded_wireframe_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1101
mapping table 586, 919
Geometry
application assertion 241, 264, 285, 299
application object 128
ARM diagrams 1301
mapping table 576, 863, 893
get_basis_surface
AIM EXPRESS listing functions 1185
get_description_value
AIM EXPRESS listing functions 1186
get_id_value
AIM EXPRESS listing functions 1186
get_role
AIM EXPRESS listing functions 1186, 1187
global_uncertainty_assigned_context
AIM diagrams 1368
AIM EXPRESS listing entities 1102
global_unit_assigned_context
AIM diagrams 1338
AIM EXPRESS listing entities 1102
government peculiar drawings
definition 11

group
AIM diagrams . . . 1323
AIM EXPRESS listing entities . . . 1102
Group_parent_column_header
application assertion . . . 269, 306
application object . . . 128
ARM diagrams . . . 1299
mapping table . . . 863
Group_parent_content_holder
application assertion . . . 269, 306
application object . . . 129
ARM diagrams . . . 1299
mapping table . . . 864
Group_parent_header
application assertion . . . 306
application object . . . 129
ARM diagrams . . . 1299
mapping table . . . 864
group_relationship
AIM diagrams . . . 1323
AIM EXPRESS listing entities . . . 1103
half_space_solid
AIM diagrams . . . 1357
AIM EXPRESS listing entities . . . 1103
Header
application assertion . . . 253, 257, 258, 261, 269, 270, 274, 278, 288, 290, 292
application object . . . 130
ARM diagrams . . . 1289
mapping table . . . 354
Header_block
application assertion . . . 270
application object . . . 131
ARM diagrams . . . 1299
mapping table . . . 864
header_configuration_restricts_property_definition
AIM EXPRESS listing rules . . . 1160
AIM EXPRESS short listing rules . . . 1000
Header_configuration_with_element_identification
application assertion . . . 269, 270, 292, 293
application object . . . 132
ARM diagrams . . . 1289
mapping table . . . 357
Header_presentation
application assertion . . . 270, 271, 284
application object . . . 133
ARM diagrams . . . 1299
mapping table . . . 865
hour_in_day
AIM EXPRESS listing types . . . 1051

hyperbola
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1103
ICAM
definition . . . 14
ID
definition . . . 14
id_attribute
AIM diagrams . . . 1342
AIM EXPRESS listing entities . . . 1103
id_attribute_select
AIM EXPRESS listing types . . . 1051
IDEF0
definition . . . 14
identification_assignment
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1103
identification_assignment_relationship
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1103
identification_item
AIM EXPRESS listing types . . . 1052
AIM EXPRESS short listing types . . . 957
identification_of_sheet_constraint
AIM EXPRESS listing rules . . . 1161
AIM EXPRESS short listing rules . . . 1001
identification_role
AIM diagrams . . . 1343
AIM EXPRESS listing entities . . . 1103
Identifier
application assertion . 239, 240, 244, 245, 247, 248, 250, 257, 259, 261-264, 266, 267, 272, 276, 278, 280, 283, 285-287, 290, 291, 294, 296-298, 301, 302, 305, 307
application object . . . 133
ARM diagrams . . . 1305
mapping table . . . 358, 865
identifier
AIM EXPRESS listing types . . . 1052
mapping table . . . 358
IDL
definition . . . 15
IGES
definition . . . 15
IL
definition . . . 15
in-house documents
definition . . . 11
indentured data list
definition . . . 12

Indentured_data_list
application assertion . . . 271
application object . . . 134
ARM diagrams . . . 1284
mapping table . . . 618, 698
Unit of Functionality . . . 36
Indentured_data_list_body
application assertion . . . 271
application object . . . 134
ARM diagrams . . . 1284
mapping table . . . 618
Indentured_data_list_entry
application assertion . . . 271-275
application object . . . 135
ARM diagrams . . . 1284
mapping table . . . 623
Indentured_data_list_header
application assertion . . . 271, 274, 275
application object . . . 139
ARM diagrams . . . 1284
mapping table . . . 639
indentured_data_list_identification_constraint
AIM EXPRESS listing rules . . . 1161
AIM EXPRESS short listing rules . . . 1002
Indentured_data_list_tabulation
application assertion . . . 271, 275
application object . . . 140
ARM diagrams . . . 1284
mapping table . . . 640
indentured_level_tag_identification_constraint
AIM EXPRESS listing rules . . . 1162
AIM EXPRESS short listing rules . . . 1003
Indentured_list_by_document
application assertion . . . 256, 275
application object . . . 141
ARM diagrams . . . 1276
mapping table . . . 469, 643
Indentured_list_by_part
application assertion . . . 256, 275, 276
application object . . . 142
ARM diagrams . . . 1276
mapping table . . . 470, 644
Indentured_list_by_part_with_document_references_to_parts
application assertion . . . 256, 275, 276
application object . . . 143
ARM diagrams . . . 1276
mapping table . . . 472, 645

indentured_list_method_identification_constraint
AIM EXPRESS listing rules 1162
AIM EXPRESS short listing rules 1004
Independent_property
application assertion 276, 277
application object 145
ARM diagrams 1302
mapping table 474, 647, 781, 893
Independent_property_relationship
application assertion 277
application object 146
ARM diagrams 1302
mapping table 475, 648, 782, 894
Independent_property_usage
application assertion 277
application object 147
ARM diagrams 1302
mapping table 476, 649, 783, 895
index list
definition 12
Index_list
application assertion 277
application object 147
ARM diagrams 1285
mapping table 699
Unit of Functionality 40
Index_list_body
application assertion 277, 278
application object 148
ARM diagrams 1285
mapping table 700
Index_list_entry
application assertion 277, 278
application object 149
ARM diagrams 1285
mapping table 702
Index_list_header
application assertion 277, 278
application object 150
ARM diagrams 1285
mapping table 703
industry standard drawing
definition 12
integer
mapping table 317, 853
intelligent data format
definition 12

intersection_curve
AIM diagrams 1350
AIM EXPRESS listing entities 1103
invisibility
AIM diagrams 1319
AIM EXPRESS listing entities 1103
invisible_item
AIM EXPRESS listing types 1052
Item
application assertion 241, 268, 278-280, 283, 286, 303, 307
application object 150
ARM diagrams 1288
mapping table 358
item_defined_transformation
AIM diagrams 1312
AIM EXPRESS listing entities 1104
Item_identification
application assertion 249, 253, 258, 272, 275, 279-282, 291, 295, 307, 308
application object 153
ARM diagrams 1287
mapping table 476, 540, 577, 649, 704, 722, 784, 896, 931
item_identified_representation_usage
AIM diagrams 1312
AIM EXPRESS listing entities 1104
item_in_context
AIM EXPRESS listing functions 1187
Item_list
application assertion 281, 290
application object 155
ARM diagrams 1281
mapping table 789
Item_list_drawing
application assertion 281, 282
application object 158
ARM diagrams 1281
mapping table 791
Item_parent_to_item_child_relationship
application assertion 255, 272, 282
application object 159
ARM diagrams 1277
mapping table 482, 654
Item_parent_to_tdp_element_child_relationship
application assertion 255, 273, 282
application object 160
ARM diagrams 1277
mapping table 482, 654
item_source_information_identification_constraint
AIM EXPRESS listing rules 1163
AIM EXPRESS short listing rules 1005

Item_type
application assertion 281-283
application object 160
ARM diagrams 1293
mapping table 484, 545, 582, 656, 792, 901
Item_usage
application assertion 241, 281, 283
application object 161
ARM diagrams 1281
mapping table 793
knot_type
AIM EXPRESS listing types 1052
Label
application assertion 240, 241, 243, 245, 246, 250, 255, 260, 261, 263, 265, 266, 268, 269, 273, 277, 279, 281, 282, 284, 287, 288, 293, 294, 296, 298-302, 309
application object 163
ARM diagrams 1304
mapping table 371, 865
label
AIM EXPRESS listing types 1052
language_assignment
AIM diagrams 1344
AIM EXPRESS listing entities 1104
AIM EXPRESS short listing entities 973
language_item
AIM EXPRESS listing types 1052
AIM EXPRESS short listing types 958
layered_item
AIM EXPRESS listing types 1052
leap_year
AIM EXPRESS listing functions 1187
length_measure
AIM EXPRESS listing types 1052
length_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1104
length_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1104
library_context
AIM diagrams 1307
AIM EXPRESS listing entities 1104
light_source
AIM EXPRESS listing entities 1104
light_source_ambient
AIM diagrams 1362
AIM EXPRESS listing entities 1105

light_source_directional
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1105
light_source_positional
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1105
light_source_spot
AIM diagrams . . . 1362
AIM EXPRESS listing entities . . . 1105
line
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1105
list_face_loops
AIM EXPRESS listing functions . . . 1188
List_item_usage
application assertion . . . 283, 284
application object . . . 164
ARM diagrams . . . 1280
mapping table . . . 796
list_of_reversible_topology_item
AIM EXPRESS listing types . . . 1052
list_of_topology_reversed
AIM EXPRESS listing functions . . . 1188
List_presentation
application assertion . . . 242, 252, 284, 285
application object . . . 166
ARM diagrams . . . 1298
mapping table . . . 865
list_representation_item
AIM EXPRESS listing types . . . 1052
mapping table . . . 746
list_to_array
AIM EXPRESS listing functions . . . 1188
list_to_set
AIM EXPRESS listing functions . . . 1189
Load_point
application assertion . . . 265, 285, 305
application object . . . 167
ARM diagrams . . . 1300
mapping table . . . 868
local_time
AIM diagrams . . . 1337
AIM EXPRESS listing entities . . . 1105
mapping table . . . 387, 388, 873
Locally_defined_item
application object . . . 167
ARM diagrams . . . 1281
mapping table . . . 802

Location_instance
application assertion . . . 285, 289
application object . . . 168
ARM diagrams . . . 1282
mapping table . . . 803
loop
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1105
Lot_effectivity
application assertion . . . 285, 286
application object . . . 169
ARM diagrams . . . 1294
mapping table . . . 484, 657, 805
lot_effectivity
AIM diagrams . . . 1328
AIM EXPRESS listing entities . . . 1105
mapping table . . . 450, 484, 610, 657, 762, 805
luminous_intensity_measure
AIM EXPRESS listing types . . . 1052
luminous_intensity_measure_with_unit
AIM diagrams . . . 1340
AIM EXPRESS listing entities . . . 1106
luminous_intensity_unit
AIM diagrams . . . 1338
AIM EXPRESS listing entities . . . 1106
Made_from_stock_material
application assertion . . . 286
application object . . . 170
ARM diagrams . . . 1281
mapping table . . . 806
make_array_of_array
AIM EXPRESS listing functions . . . 1189
Make_from_relationship
application assertion . . . 286
application object . . . 171
ARM diagrams . . . 1282
mapping table . . . 808
make_from_usage_option
AIM diagrams . . . 1310
AIM EXPRESS listing entities . . . 1106
mapping table . . . 806, 808
make_from_usage_option_with_reference_designator
AIM diagrams . . . 1310
AIM EXPRESS listing entities . . . 1106
AIM EXPRESS short listing entities . . . 974
mapping table . . . 806, 808
manifold_solid_brep
AIM diagrams . . . 1355
AIM EXPRESS listing entities . . . 1106

manifold_surface_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1106
mapping table 586, 919
mapped_item
AIM diagrams 1314
AIM EXPRESS listing entities 1109
marker_select
AIM EXPRESS listing types 1053
marker_type
AIM EXPRESS listing types 1053
mass_measure
AIM EXPRESS listing types 1053
mass_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1109
mass_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1109
material_designation
AIM diagrams 1312
AIM EXPRESS listing entities 1110
mapping table 842
Measure
application assertion 276, 284, 286, 287, 298, 309
application object 172
ARM diagrams 1304
mapping table 371, 485, 657, 809, 868
measure_qualification
AIM diagrams 1340
AIM EXPRESS listing entities 1110
measure_representation_item
AIM diagrams 1340
AIM EXPRESS listing entities 1110
mapping table 844
measure_value
AIM EXPRESS listing types 1053
measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1110
mapping table 488, 492, 515, 660, 671, 691, 812, 829, 850
mechanical_design_geometric_presentation_area
AIM diagrams 1320
AIM EXPRESS listing entities 1110
mechanical_design_geometric_presentation_representation
AIM diagrams 1313
AIM EXPRESS listing entities 1111
mapping table 911, 915

mechanical_design_shaded_presentation_area
AIM diagrams 1320
AIM EXPRESS listing entities 1115
mechanical_design_shaded_presentation_representation
AIM diagrams 1313
AIM EXPRESS listing entities 1116
mapping table 583
message
AIM EXPRESS listing types 1053
minute_in_hour
AIM EXPRESS listing types 1053
mixed_loop_type_set
AIM EXPRESS listing functions 1189
month_in_year_number
AIM EXPRESS listing types 1053
msb_shells
AIM EXPRESS listing functions 1190
msf_curve_check
AIM EXPRESS listing functions 1190
msf_surface_check
AIM EXPRESS listing functions 1191
name_attribute
AIM diagrams 1371
AIM EXPRESS listing entities 1121
name_attribute_select
AIM EXPRESS listing types 1053
named_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1121
mapping table 371, 485, 657, 809, 868
next_assembly_usage_occurrence
AIM diagrams 1310
AIM EXPRESS listing entities 1122
mapping table 757
nmsf_curve_check
AIM EXPRESS listing functions 1192
nmsf_surface_check
AIM EXPRESS listing functions 1193
non_manifold_surface_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1122
mapping table 586, 919
normalise
AIM EXPRESS listing functions 1193
Notation
application assertion 249, 251, 252, 257, 266, 271, 273, 278, 279, 284, 287, 288, 290, 297
application object 172
ARM diagrams 1296
mapping table 485, 546, 657, 709, 727, 809

notation_type_identification_constraint
AIM EXPRESS listing rules 1164
AIM EXPRESS short listing rules 1006
number
mapping table 853
Number_with_units
application assertion 286, 287, 296, 304, 309
application object 174
ARM diagrams 1282
mapping table 488, 660, 812
numeric_measure
AIM EXPRESS listing types 1054
object_role
AIM diagrams 1342
AIM EXPRESS listing entities 1125
offset_curve_2d
AIM diagrams 1347
AIM EXPRESS listing entities 1125
offset_curve_3d
AIM diagrams 1347
AIM EXPRESS listing entities 1125
offset_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1125
OL
definition 15
open_shell
AIM diagrams 1355
AIM EXPRESS listing entities 1125
open_shell_reversed
AIM EXPRESS listing functions 1194
organization
AIM diagrams 1333
AIM EXPRESS listing entities 1125
mapping table 331, 332, 511
organization_assignment
AIM diagrams 1334
AIM EXPRESS listing entities 1125
mapping table 345
organization_item
AIM EXPRESS listing types 1054
AIM EXPRESS short listing types 958
organization_relationship
AIM diagrams 1333
AIM EXPRESS listing entities 1125
organization_role
AIM diagrams 1334
AIM EXPRESS listing entities 1126

organizational_address
AIM diagrams 1333
AIM EXPRESS listing entities 1126
organizational_project
AIM diagrams 1335
AIM EXPRESS listing entities 1126
mapping table 488, 490, 491, 668-670, 826-828
organizational_project_assignment
AIM diagrams 1335
AIM EXPRESS listing entities 1126
organizational_project_item
AIM EXPRESS listing types 1054
AIM EXPRESS short listing types 958
organizational_project_relationship
AIM diagrams 1335
AIM EXPRESS listing entities 1126
organizational_project_role
AIM diagrams 1335
AIM EXPRESS listing entities 1126
oriented_closed_shell
AIM diagrams 1355
AIM EXPRESS listing entities 1126
oriented_edge
AIM diagrams 1354
AIM EXPRESS listing entities 1127
oriented_face
AIM diagrams 1354
AIM EXPRESS listing entities 1127
oriented_open_shell
AIM diagrams 1355
AIM EXPRESS listing entities 1127
oriented_path
AIM diagrams 1354
AIM EXPRESS listing entities 1127
oriented_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1127
orthogonal_complement
AIM EXPRESS listing functions 1195
Other_list
application assertion 287
application object 175
ARM diagrams 1286
mapping table 549, 730
Unit of Functionality 42

Other_list_body
application assertion . . . 287, 288
application object . . . 175
ARM diagrams . . . 1286
mapping table . . . 731
Other_list_body_table
application assertion . . . 288
application object . . . 176
ARM diagrams . . . 1286
mapping table . . . 733
Other_list_header
application assertion . . . 287, 288
application object . . . 177
ARM diagrams . . . 1286
mapping table . . . 550, 734
other_list_table_representation
AIM diagrams . . . 1313
AIM EXPRESS listing entities . . . 1128
AIM EXPRESS short listing entities . . . 974
mapping table . . . 733
Other_reference_document
application assertion . . . 240, 244, 262, 297
application object . . . 178
ARM diagrams . . . 1303
mapping table . . . 372
outer_boundary_curve
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1128
over_riding_styled_item
AIM EXPRESS listing entities . . . 1128
Page_parameters
application assertion . . . 288, 289
application object . . . 178
ARM diagrams . . . 1298
mapping table . . . 869
Page_presentation
application assertion . . . 285, 289
application object . . . 180
ARM diagrams . . . 1298
mapping table . . . 871
parabola
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1128
parameter_value
AIM EXPRESS listing types . . . 1054
parametric_representation_context
AIM diagrams . . . 1313
AIM EXPRESS listing entities . . . 1128

Part_occurrence_in_assembly
application assertion . . . 289
application object . . . 181
ARM diagrams . . . 1282
mapping table . . . 812
parts list
definition . . . 12
Parts_list
application assertion . . . 289
application object . . . 182
ARM diagrams . . . 1280
mapping table . . . 550, 817
Unit of Functionality . . . 45
Parts_list_body
application assertion . . . 289, 290
application object . . . 182
ARM diagrams . . . 1280
mapping table . . . 818
Parts_list_header
application assertion . . . 289, 290
application object . . . 184
ARM diagrams . . . 1280
mapping table . . . 551, 821
path
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1128
path information
definition . . . 12
path_head_to_tail
AIM EXPRESS listing functions . . . 1195
path_reversed
AIM EXPRESS listing functions . . . 1195
pcurve
AIM diagrams . . . 1347
AIM EXPRESS listing entities . . . 1128
pcurve_or_surface
AIM EXPRESS listing types . . . 1054
PDM
definition . . . 15
PDS
definition . . . 15
Person
application assertion . . . 260, 290, 291, 297, 306
application object . . . 184
ARM diagrams . . . 1297
mapping table . . . 372

person
AIM diagrams . . . 1333
AIM EXPRESS listing entities . . . 1128
mapping table . . . 372
Person_and_organization
application assertion . . . 242, 244, 247, 260, 264, 267, 291, 292, 294, 298
application object . . . 185
ARM diagrams . . . 1297
mapping table . . . 373
person_and_organization
AIM diagrams . . . 1332
AIM EXPRESS listing entities . . . 1128
mapping table . . . 511
person_and_organization_assignment
AIM diagrams . . . 1332
AIM EXPRESS listing entities . . . 1129
mapping table . . . 345
person_and_organization_item
AIM EXPRESS listing types . . . 1054
AIM EXPRESS short listing types . . . 959
person_and_organization_role
AIM diagrams . . . 1332
AIM EXPRESS listing entities . . . 1129
person_organization_select
AIM EXPRESS listing types . . . 1055
mapping table . . . 373
personal_address
AIM diagrams . . . 1333
AIM EXPRESS listing entities . . . 1129
PICS
definition . . . 15
PL
definition . . . 15
placement
AIM diagrams . . . 1345
AIM EXPRESS listing entities . . . 1129
planar_box
AIM diagrams . . . 1317
AIM EXPRESS listing entities . . . 1129
planar_extent
AIM diagrams . . . 1317
AIM EXPRESS listing entities . . . 1129
plane
AIM diagrams . . . 1351
AIM EXPRESS listing entities . . . 1129
plane_angle_measure
AIM EXPRESS listing types . . . 1055

plane_angle_measure_with_unit
AIM diagrams . . . 1340
AIM EXPRESS listing entities . . . 1129
plane_angle_unit
AIM diagrams . . . 1338
AIM EXPRESS listing entities . . . 1130
point
AIM diagrams . . . 1346
AIM EXPRESS listing entities . . . 1130
point_on_curve
AIM diagrams . . . 1346
AIM EXPRESS listing entities . . . 1130
point_on_surface
AIM diagrams . . . 1346
AIM EXPRESS listing entities . . . 1130
point_replica
AIM diagrams . . . 1346
AIM EXPRESS listing entities . . . 1130
point_style
AIM diagrams . . . 1319
AIM EXPRESS listing entities . . . 1130
poly_loop
AIM diagrams . . . 1354
AIM EXPRESS listing entities . . . 1130
polyline
AIM diagrams . . . 1348
AIM EXPRESS listing entities . . . 1130
positive_length_measure
AIM EXPRESS listing types . . . 1055
positive_plane_angle_measure
AIM EXPRESS listing types . . . 1055
positive_ratio_measure
AIM EXPRESS listing types . . . 1055
pre_defined_character_glyph
AIM diagrams . . . 1318
AIM EXPRESS listing entities . . . 1131
pre_defined_colour
AIM diagrams . . . 1360
AIM EXPRESS listing entities . . . 1131
pre_defined_curve_font
AIM diagrams . . . 1360
AIM EXPRESS listing entities . . . 1131
pre_defined_item
AIM diagrams . . . 1360
AIM EXPRESS listing entities . . . 1131
pre_defined_text_font
AIM diagrams . . . 1360
AIM EXPRESS listing entities . . . 1131

precision_qualifier
AIM diagrams . . . 1340
AIM EXPRESS listing entities . . . 1131
preferred_surface_curve_representation
AIM EXPRESS listing types . . . 1055
PRES
definition . . . 15
presentable_text
AIM EXPRESS listing types . . . 1055
Presentation
Unit of Functionality . . . 49
presentation_area
AIM diagrams . . . 1320
AIM EXPRESS listing entities . . . 1131
mapping table . . . 871
presentation_representation
AIM diagrams . . . 1320
AIM EXPRESS listing entities . . . 1131
mapping table . . . 583, 911, 915
presentation_representation_select
AIM EXPRESS listing types . . . 1055
presentation_set
AIM diagrams . . . 1320
AIM EXPRESS listing entities . . . 1131
mapping table . . . 865
presentation_size
AIM diagrams . . . 1317
AIM EXPRESS listing entities . . . 1131
mapping table . . . 869
presentation_size_assignment_select
AIM EXPRESS listing types . . . 1055
presentation_style_assignment
AIM diagrams . . . 1319
AIM EXPRESS listing entities . . . 1132
presentation_style_by_context
AIM diagrams . . . 1350
AIM EXPRESS listing entities . . . 1132
presentation_style_select
AIM EXPRESS listing types . . . 1055
presentation_view
AIM diagrams . . . 1320
AIM EXPRESS listing entities . . . 1132
presented_item
AIM diagrams . . . 1315
AIM EXPRESS listing entities . . . 1132
presented_item_item
AIM EXPRESS listing types . . . 1055
AIM EXPRESS short listing types . . . 959

presented_item_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1132
process_product_association
AIM diagrams 1321
AIM EXPRESS listing entities 1132
product
AIM diagrams 1308
AIM EXPRESS listing entities 1132
mapping table 394, 448, 539, 598, 609, 753, 761, 882
product data set
definition 12
product drawings and associated lists
definition 13
product information
definition 13
product_category
AIM diagrams 1308
AIM EXPRESS listing entities 1132
mapping table 354, 484, 545, 582, 656, 792, 901
product_category_graph_members
AIM EXPRESS listing functions 1195
AIM EXPRESS short listing functions 1017
product_category_relationship
AIM diagrams 1308
AIM EXPRESS listing entities 1133
product_concept
AIM diagrams 1307
AIM EXPRESS listing entities 1133
mapping table 376
product_concept_context
AIM diagrams 1307
AIM EXPRESS listing entities 1133
Product_configuration
application assertion 246, 261, 291, 292
application object 186
ARM diagrams 1294
mapping table 374
product_context
AIM diagrams 1307
AIM EXPRESS listing entities 1133
Product_data_set
application assertion 242, 292
application object 188
ARM diagrams 1301
mapping table 551, 660, 735, 822, 902
Unit of Functionality 51

Product_data_set_with_format
application assertion 292
application object 189
ARM diagrams 1301
mapping table 552, 664, 736, 823, 906
Product_data_set_with_shading
application assertion 292, 293
application object 189
ARM diagrams 1301
mapping table 553, 665, 737, 824, 907
Product_data_set_without_format
application assertion 293
application object 190
ARM diagrams 1301
mapping table 554, 666, 738, 825, 909
product_definition
AIM diagrams 1309
AIM EXPRESS listing entities 1133
mapping table . 402, 413, 430, 440, 513, 528, 530, 536, 576, 618, 623, 688, 689, 696, 700, 702, 731, 789, 818, 847
product_definition_context
AIM diagrams 1307
AIM EXPRESS listing entities 1133
product_definition_context_association
AIM diagrams 1309
AIM EXPRESS listing entities 1133
product_definition_context_role
AIM diagrams 1309
AIM EXPRESS listing entities 1133
product_definition_effectivity
AIM diagrams 1328
AIM EXPRESS listing entities 1133
product_definition_formation
AIM diagrams 1308
AIM EXPRESS listing entities 1134
mapping table . 312, 348, 372, 376, 389, 429, 439, 448, 476, 501, 507, 508, 511, 519, 523, 527, 537, 538, 540, 549-554, 560, 562-566, 571, 577, 590, 593, 594, 602, 608, 618, 649, 660, 664-666, 681, 685, 686, 694, 696-699, 704, 717-719, 721, 722, 730, 735-738, 744, 747, 749, 757, 760, 784, 791, 817, 822-825, 838, 841, 842, 846, 874, 881, 896, 902, 906, 907, 909
product_definition_formation_relationship
AIM diagrams 1308
AIM EXPRESS listing entities 1134
mapping table 385, 511, 512, 654
product_definition_formation_with_specified_source
AIM diagrams 1308
AIM EXPRESS listing entities 1134

product_definition_occurrence_relationship
AIM diagrams 1309
AIM EXPRESS listing entities 1134
mapping table 482
product_definition_process
AIM diagrams 1321
AIM EXPRESS listing entities 1134
product_definition_relationship
AIM diagrams 1310
AIM EXPRESS listing entities 1134
mapping table 385, 687, 688, 780, 781, 793, 796, 802
product_definition_shape
AIM diagrams 1311
AIM EXPRESS listing entities 1134
product_definition_substitute
AIM diagrams 1310
AIM EXPRESS listing entities 1135
product_definition_usage
AIM diagrams 1310
AIM EXPRESS listing entities 1135
product_definition_with_associated_documents
AIM diagrams 1309
AIM EXPRESS listing entities 1135
mapping table 385, 402, 413, 440, 623
Product_model
application assertion 292, 293
application object 191
ARM diagrams 1294
mapping table 376
product_or_formation_or_definition
AIM EXPRESS listing types 1056
Product_presentation
application assertion 292, 293, 300
application object 191
ARM diagrams 1301
mapping table 583, 911
product_related_product_category
AIM diagrams 1308
AIM EXPRESS listing entities 1135
product_relationship
AIM diagrams 1334
AIM EXPRESS listing entities 1135
product_requires_category
AIM EXPRESS listing rules 1164
AIM EXPRESS short listing rules 1007
product_requires_version
AIM EXPRESS listing rules 1164
AIM EXPRESS short listing rules 1007

product_version_requires_person_organization
AIM EXPRESS listing rules 1164
AIM EXPRESS short listing functions 1008
Project
application assertion 264, 293, 294
application object 192
ARM diagrams 1297
mapping table 488, 668, 826
Promissory_usage
application object 193
ARM diagrams 1282
mapping table 828
promissory_usage_occurrence
AIM diagrams 1310
AIM EXPRESS listing entities 1135
mapping table 757, 828
Property_assignment
application assertion 277, 294
application object 194
ARM diagrams 1302
mapping table 491, 670, 828, 914
property_definition
AIM diagrams 1311
AIM EXPRESS listing entities 1135
mapping table . . 335, 354, 357, 358, 435, 491-493, 536, 550, 551, 639, 670, 671, 703, 734, 821, 828, 829, 914
property_definition_relationship
AIM diagrams 1311
AIM EXPRESS listing entities 1136
property_definition_representation
AIM diagrams 1311
AIM EXPRESS listing entities 1136
qualified_representation_item
AIM diagrams 1314
AIM EXPRESS listing entities 1136
quantified_assembly_component_usage
AIM diagrams 1310
AIM EXPRESS listing entities 1136
mapping table 757
Quantified_part_usage_in_assembly
application assertion 302
application object 194
ARM diagrams 1282
mapping table 829
Quantity
application assertion 283, 294, 295, 299
application object 195
ARM diagrams 1281
mapping table 671, 829

quasi_uniform_curve
AIM diagrams 1349
AIM EXPRESS listing entities 1136
quasi_uniform_surface
AIM diagrams 1353
AIM EXPRESS listing entities 1136
ratio_measure
AIM EXPRESS listing types 1056
ratio_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1136
ratio_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1136
rational_b_spline_curve
AIM diagrams 1349
AIM EXPRESS listing entities 1137
rational_b_spline_surface
AIM diagrams 1353
AIM EXPRESS listing entities 1137
RD
definition 15
real
mapping table 853
Reason
application assertion 253, 295
application object 196
ARM diagrams 1274
mapping table 493
rectangular_composite_surface
AIM diagrams 1352
AIM EXPRESS listing entities 1137
rectangular_trimmed_surface
AIM diagrams 1352
AIM EXPRESS listing entities 1137
reference document
definition 13
Reference_document
application assertion 249, 251, 258, 273, 279, 295, 296
application object 199
ARM diagrams 1303
mapping table 376
Unit of Functionality 54
reference_document_requires_subcategorization
AIM EXPRESS listing rules 1165
AIM EXPRESS short listing rules 1009

Reference_document_usage
application assertion 279, 286, 296
application object 200
ARM diagrams 1303
mapping table 376
Relative_event
application assertion 296
application object 201
ARM diagrams 1297
mapping table 494, 672, 830
relative_event_occurrence
AIM diagrams 1365
AIM EXPRESS listing entities 1138
mapping table 494, 672, 830
Release_authentication
application assertion 246, 296, 297
application object 202
ARM diagrams 1290
mapping table 378
release_authentication_string_restriction
AIM EXPRESS listing rules 1166
AIM EXPRESS short listing rules 1010
rendering_properties_select
AIM EXPRESS listing types 1056
reparametrised_composite_curve_segment
AIM diagrams 1348
AIM EXPRESS listing entities 1138
representation
AIM diagrams 1313
AIM EXPRESS listing entities 1138
mapping table . 383, 384, 398, 466, 504, 505, 508, 604, 683, 776, 840, 843, 862, 879, 889, 892, 893
representation_context
AIM diagrams 1313
AIM EXPRESS listing entities 1138
representation_item
AIM diagrams 1314
AIM EXPRESS listing entities 1138
mapping table 516, 691, 850
representation_item_relationship
AIM diagrams 1314
AIM EXPRESS listing entities 1138
representation_map
AIM diagrams 1314
AIM EXPRESS listing entities 1139
representation_relationship
AIM diagrams 1367
AIM EXPRESS listing entities 1139

representation_relationship_with_transformation
AIM diagrams 1368
AIM EXPRESS listing entities 1139
represented_definition
AIM EXPRESS listing types 1056
Retrofit_usage
application assertion 273, 283, 297
application object 203
ARM diagrams 1280
mapping table 672, 830
reversible_topology
AIM EXPRESS listing types 1056
reversible_topology_item
AIM EXPRESS listing types 1056
Revision
application assertion 252, 257, 271, 278, 288, 290, 297, 298
application object 204
ARM diagrams 1280
mapping table 495, 555, 675, 712, 739, 832
revolved_face_solid
AIM diagrams 1355
AIM EXPRESS listing entities 1139
right_angular_wedge
AIM diagrams 1358
AIM EXPRESS listing entities 1139
right_circular_cone
AIM diagrams 1358
AIM EXPRESS listing entities 1139
right_circular_cylinder
AIM diagrams 1358
AIM EXPRESS listing entities 1139
role_association
AIM diagrams 1342
AIM EXPRESS listing entities 1140
role_select
AIM EXPRESS listing types 1056
scalar_times_vector
AIM EXPRESS listing functions 1196
seam_curve
AIM diagrams 1350
AIM EXPRESS listing entities 1140
second_in_minute
AIM EXPRESS listing types 1056
second_proj_axis
AIM EXPRESS listing functions 1196

Security_classification
application assertion 246, 267, 284, 298
application object 205
ARM diagrams 1290
mapping table 380
security_classification
AIM diagrams 1331
AIM EXPRESS listing entities 1140
mapping table 380
security_classification_assignment
AIM diagrams 1331
AIM EXPRESS listing entities 1140
security_classification_date_string_restriction
AIM EXPRESS listing rules 1167
AIM EXPRESS short listing rules 1012
security_classification_item
AIM EXPRESS listing types 1056
AIM EXPRESS short listing types 960
security_classification_level
AIM diagrams 1331
AIM EXPRESS listing entities 1140
Sequence_effectivity
application assertion 298, 299
application object 207
ARM diagrams 1294
mapping table 492, 500, 680, 837
serial_numbered_effectivity
AIM diagrams 1328
AIM EXPRESS listing entities 1140
mapping table 450, 500, 610, 680, 762, 837
set_of_reversible_topology_item
AIM EXPRESS listing types 1057
set_of_topology_reversed
AIM EXPRESS listing functions 1197
Shaded_shape_model
application assertion 293, 299
application object 208
ARM diagrams 1301
mapping table 915
shading_curve_method
AIM EXPRESS listing types 1057
shading_surface_method
AIM EXPRESS listing types 1057
shape_aspect
AIM diagrams 1312
AIM EXPRESS listing entities 1140
shape_aspect_relationship
AIM diagrams 1312
AIM EXPRESS listing entities 1141

shape_definition
AIM EXPRESS listing types 1057
shape_definition_representation
AIM diagrams 1368
AIM EXPRESS listing entities 1141
Shape_model
application assertion 279, 293, 299
application object 208
ARM diagrams 1301
mapping table 586, 919
shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1141
mapping table 586, 919
shape_representation_relationship
AIM diagrams 1367
AIM EXPRESS listing entities 1141
Sheet
application assertion 258, 261, 299, 300
application object 209
ARM diagrams 1301
mapping table 501, 560, 590, 681, 744, 838, 923
shell
AIM EXPRESS listing types 1057
shell_based_surface_model
AIM diagrams 1356
AIM EXPRESS listing entities 1141
shell_based_wireframe_model
AIM diagrams 1356
AIM EXPRESS listing entities 1141
shell_based_wireframe_shape_representation
AIM diagrams 1315
AIM EXPRESS listing entities 1141
mapping table 586, 919
shell_reversed
AIM EXPRESS listing functions 1197
si_prefix
AIM EXPRESS listing types 1057
si_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1143
si_unit_name
AIM EXPRESS listing types 1057
Simple_list_of_elements
application assertion 252, 300
application object 210
ARM diagrams 1275
mapping table 503

simple_list_of_elements_constraint
AIM EXPRESS listing rules 1167
AIM EXPRESS short listing rules 1013
Simple_list_of_files
application assertion 253, 300
application object 210
ARM diagrams 1275
mapping table 503
simple_list_of_files_constraint
AIM EXPRESS listing rules 1168
AIM EXPRESS short listing rules 1014
Single_document_list
application object 211
ARM diagrams 1273
mapping table 562, 717
Size_characteristics
application assertion 251, 267, 300
application object 211
ARM diagrams 1279
mapping table 383
Size_characteristics_full_size
application assertion 269
application object 212
ARM diagrams 1279
mapping table 384
Size_characteristics_internal_divisions
application assertion 300, 301
application object 212
ARM diagrams 1279
mapping table 504, 925
Size_characteristics_sheet_across_file
application assertion 301
application object 213
ARM diagrams 1279
mapping table 504, 925
size_select
AIM EXPRESS listing types 1058
solid_angle_measure
AIM EXPRESS listing types 1058
solid_angle_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1144
solid_angle_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1144
solid_model
AIM diagrams 1355
AIM EXPRESS listing entities 1144

solid_replica
AIM diagrams 1355
AIM EXPRESS listing entities 1144
source
AIM EXPRESS listing types 1058
Source_file
application assertion 292
application object 214
ARM diagrams 1278
mapping table 926
Source_information_type
application assertion 241, 281, 301
application object 214
ARM diagrams 1296
mapping table 505, 683, 840
source_item
AIM EXPRESS listing types 1058
Source_location
application assertion 249, 251, 258, 273, 302
application object 215
ARM diagrams 1279
mapping table 506, 562, 683
special inspection equipment drawings and associated lists
definition 13
special tooling drawings and associated lists
definition 13
Special_condition
application assertion 249, 251, 258, 273, 279, 283, 302
application object 216
ARM diagrams 1296
mapping table 506, 562, 684, 840, 926
Specification_document
application object 217
ARM diagrams 1303
mapping table 507, 563, 685, 841, 936
specified_higher_usage_occurrence
AIM diagrams 1310
AIM EXPRESS listing entities 1144
mapping table 757, 841, 842
Specified_part_in_assembly_tree
application assertion 302
application object 217
ARM diagrams 1282
mapping table 841
sphere
AIM diagrams 1357
AIM EXPRESS listing entities 1145

spherical_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1145
standard parts
definition 13
Standard_document
application object 218
ARM diagrams 1303
mapping table 508, 564, 685, 842, 937
Status
application assertion 244, 302, 303
application object 218
ARM diagrams 1291
mapping table 384
Stock_material
application assertion 286, 303
application object 219
ARM diagrams 1281
mapping table 842
Stock_size
application assertion 286, 303
application object 220
ARM diagrams 1282
mapping table 843
Stock_size_parameters
application assertion 303, 304
application object 223
ARM diagrams 1282
mapping table 844
String_location_definition
application assertion 305
application object 226
ARM diagrams 1300
mapping table 871
style_context_select
AIM EXPRESS listing types 1058
styled_item
AIM diagrams 1319
AIM EXPRESS listing entities 1145
supported_item
AIM EXPRESS listing types 1058
surface
AIM diagrams 1351
AIM EXPRESS listing entities 1145
surface_boundary
AIM EXPRESS listing types 1059
surface_curve
AIM diagrams 1350
AIM EXPRESS listing entities 1145

surface_model
AIM EXPRESS listing types 1059
surface_of_linear_extrusion
AIM diagrams 1351
AIM EXPRESS listing entities 1145
surface_of_revolution
AIM diagrams 1351
AIM EXPRESS listing entities 1146
surface_patch
AIM diagrams 1352
AIM EXPRESS listing entities 1146
surface_rendering_properties
AIM diagrams 1361
AIM EXPRESS listing entities 1146
surface_replica
AIM diagrams 1351
AIM EXPRESS listing entities 1146
surface_side
AIM EXPRESS listing types 1059
surface_side_style
AIM diagrams 1359
AIM EXPRESS listing entities 1146
surface_side_style_select
AIM EXPRESS listing types 1059
surface_style_boundary
AIM diagrams 1359
AIM EXPRESS listing entities 1146
surface_style_control_grid
AIM diagrams 1359
AIM EXPRESS listing entities 1146
surface_style_element_select
AIM EXPRESS listing types 1059
surface_style_fill_area
AIM diagrams 1359
AIM EXPRESS listing entities 1147
surface_style_parameter_line
AIM diagrams 1359
AIM EXPRESS listing entities 1147
surface_style_reflectance_ambient
AIM diagrams 1360
AIM EXPRESS listing entities 1147
surface_style_reflectance_ambient_diffuse
AIM diagrams 1360
AIM EXPRESS listing entities 1147
surface_style_reflectance_ambient_diffuse_specular
AIM diagrams 1360
AIM EXPRESS listing entities 1147

surface_style_rendering
AIM diagrams . . . 1359
AIM EXPRESS listing entities . . . 1147
surface_style_rendering_with_properties
AIM diagrams . . . 1359
AIM EXPRESS listing entities . . . 1147
surface_style_segmentation_curve
AIM diagrams . . . 1359
AIM EXPRESS listing entities . . . 1147
surface_style_silhouette
AIM diagrams . . . 1359
AIM EXPRESS listing entities . . . 1147
surface_style_transparent
AIM diagrams . . . 1360
AIM EXPRESS listing entities . . . 1147
surface_style_usage
AIM diagrams . . . 1359
AIM EXPRESS listing entities . . . 1147
surface_weights_positive
AIM EXPRESS listing functions . . . 1197
swept_face_solid
AIM diagrams . . . 1355
AIM EXPRESS listing entities . . . 1148
swept_surface
AIM diagrams . . . 1351
AIM EXPRESS listing entities . . . 1148
symbol_representation
AIM diagrams . . . 1315
AIM EXPRESS listing entities . . . 1148
System_declaration
application assertion . . . 264, 267, 305
application object . . . 226
ARM diagrams . . . 1278
mapping table . . . 508, 926
System_destination
application assertion . . . 254, 305, 306
application object . . . 227
ARM diagrams . . . 1274
mapping table . . . 511
Tabulation
application assertion . . . 242, 271, 306
application object . . . 228
ARM diagrams . . . 1299
mapping table . . . 872
Tabulation_entry
application assertion . . . 288, 306
application object . . . 230
ARM diagrams . . . 1286
mapping table . . . 746

Tabulation_header
application assertion 306, 307
application object 230
ARM diagrams 1299
mapping table 872
TDP
definition 15
TDP element
definition 13
Tdp_element
application assertion 307
application object 230
ARM diagrams 1273
mapping table 511, 565, 593, 686, 718, 747, 846, 930, 937
Tdp_element_and_item_association
application assertion 307
application object 231
ARM diagrams 1273
mapping table 385
Tdp_element_list_item
application assertion 290, 307
application object 231
ARM diagrams 1280
mapping table 847
Tdp_element_parent_to_item_child_relationship
application assertion 255, 274, 307
application object 232
ARM diagrams 1277
mapping table 511, 687
Tdp_element_parent_to_tdp_element_child_relationship
application assertion 255, 274, 307, 308
application object 233
ARM diagrams 1277
mapping table 512, 688
Tdp_indentured_item
application assertion 255, 274, 276, 308
application object 233
ARM diagrams 1277
mapping table 513, 688
Tdp_indentured_tdp_element
application assertion 256, 274-276, 308
application object 234
ARM diagrams 1277
mapping table 513, 689
technical data package
definition 13

Text
application assertion . 239-243, 245, 247, 248, 251, 259-261, 263-266, 268, 270, 276, 277, 281, 283, 287, 289, 291, 294-298, 301-306, 308, 309
application object 234
ARM diagrams 1305
mapping table 387, 873
text
AIM EXPRESS listing types 1059
mapping table 387, 873
text_alignment
AIM EXPRESS listing types 1059
text_literal
AIM diagrams 1318
AIM EXPRESS listing entities 1148
text_literal_alignment_baseline_constraint
AIM EXPRESS listing rules 1169
AIM EXPRESS short listing rules 1016
text_literal_font_requires_externally_defined_text_font
AIM EXPRESS listing rules 1169
AIM EXPRESS short listing rules 1016
text_or_character
AIM EXPRESS listing types 1059
text_path
AIM EXPRESS listing types 1059
text_string_representation
AIM diagrams 1313
AIM EXPRESS listing entities 1148
text_style
AIM diagrams 1319
AIM EXPRESS listing entities 1149
text_style_for_defined_font
AIM diagrams 1319
AIM EXPRESS listing entities 1149
thermodynamic_temperature_measure
AIM EXPRESS listing types 1059
thermodynamic_temperature_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1149
thermodynamic_temperature_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1149
Time
application assertion 259, 308
application object 235
ARM diagrams 1297
mapping table 388, 873
time_interval
AIM diagrams 1366
AIM EXPRESS listing entities 1149

time_interval_based_effectivity
AIM diagrams 1366
AIM EXPRESS listing entities 1149
mapping table 450, 513, 689, 848
Time_interval_effectivity
application assertion 308, 309
application object 235
ARM diagrams 1294
mapping table 513, 689, 848
time_interval_relationship
AIM diagrams 1366
AIM EXPRESS listing entities 1149
time_interval_with_bounds
AIM diagrams 1366
AIM EXPRESS listing entities 1149
time_measure
AIM EXPRESS listing types 1059
time_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1150
time_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1150
topological_representation_item
AIM diagrams 1323
AIM EXPRESS listing entities 1150
topology_reversed
AIM EXPRESS listing functions 1198
toroidal_surface
AIM diagrams 1351
AIM EXPRESS listing entities 1150
torus
AIM diagrams 1357
AIM EXPRESS listing entities 1150
Transformation
application assertion 285
application object 237
ARM diagrams 1282
mapping table 849
transformation
AIM EXPRESS listing types 1060
transition_code
AIM EXPRESS listing types 1060
trimmed_curve
AIM diagrams 1348
AIM EXPRESS listing entities 1150
trimming_preference
AIM EXPRESS listing types 1060

trimming_select
AIM EXPRESS listing types 1060
type_qualifier
AIM diagrams 1340
AIM EXPRESS listing entities 1150
u_direction_count
AIM EXPRESS listing types 1060
uncertainty_measure_with_unit
AIM diagrams 1368
AIM EXPRESS listing entities 1151
uniform_curve
AIM diagrams 1349
AIM EXPRESS listing entities 1151
uniform_surface
AIM diagrams 1353
AIM EXPRESS listing entities 1151
unit
AIM EXPRESS listing types 1060
UoF
definition 15
using_items
AIM EXPRESS listing functions 1198
using_representations
AIM EXPRESS listing functions 1199
v_direction_count
AIM EXPRESS listing types 1060
valid_basis_curve_in_2d_wireframe
AIM EXPRESS listing functions 1199
valid_calendar_date
AIM EXPRESS listing functions 1200
valid_geometrically_bounded_wf_curve
AIM EXPRESS listing functions 1201
valid_geometrically_bounded_wf_point
AIM EXPRESS listing functions 1201
valid_measure_value
AIM EXPRESS listing functions 1202
valid_time
AIM EXPRESS listing functions 1202
valid_units
AIM EXPRESS listing functions 1202
valid_wireframe_edge_curve
AIM EXPRESS listing functions 1204
valid_wireframe_vertex_point
AIM EXPRESS listing functions 1204
value_qualifier
AIM EXPRESS listing types 1060
value_representation_item
AIM diagrams 1341
AIM EXPRESS listing entities 1151

vector
AIM diagrams 1345
AIM EXPRESS listing entities 1151
vector_difference
AIM EXPRESS listing functions 1205
vector_or_direction
AIM EXPRESS listing types 1060
versioned_action_request
AIM diagrams 1363
AIM EXPRESS listing entities 1151
versioned_action_request_requires_status
AIM EXPRESS listing rules 1169
vertex
AIM diagrams 1354
AIM EXPRESS listing entities 1151
vertex_loop
AIM diagrams 1354
AIM EXPRESS listing entities 1151
vertex_point
AIM diagrams 1354
AIM EXPRESS listing entities 1151
vertex_shell
AIM diagrams 1354
AIM EXPRESS listing entities 1151
view_volume
AIM diagrams 1317
AIM EXPRESS listing entities 1152
Volume
application assertion 295, 309
application object 237
ARM diagrams 1281
mapping table 515, 691, 850
volume_measure
AIM EXPRESS listing types 1060
volume_measure_with_unit
AIM diagrams 1340
AIM EXPRESS listing entities 1152
volume_unit
AIM diagrams 1338
AIM EXPRESS listing entities 1152
Weight
application assertion 280, 295, 309
application object 237
ARM diagrams 1288
mapping table 516, 691, 850
wire_shell
AIM diagrams 1354
AIM EXPRESS listing entities 1152

wireframe_model
AIM EXPRESS listing types 1060
year_number
AIM EXPRESS listing types 1060

ICS 25.040.40

Price based on 1546 pages