INTERNATIONAL STANDARD

ISO 15926-2

First edition
2003-12-15

Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities —

Part 2: **Data model**

Systèmes d'automatisation industrielle et intégration — Intégration de données de cycle de vie pour les industries de «process», y compris les usines de production de pétrole et de gaz —

Partie 2: Modèle de données

Reference number
ISO 15926-2:2003(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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.org
Web www.iso.org

Published in Switzerland

Contents

Figures

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 2.

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 document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 15926-2 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 International Standard is described in ISO 15926-1. A complete list of parts of ISO 15926 is available from the Internet:

`<http://www.tc184-sc4.org/titles/15926_Titles.rtf>`

Introduction

ISO 15926 is an International Standard for the representation of process plant life-cycle information. This representation is specified by a generic, conceptual data model that is suitable as the basis for implementation in a shared database or data warehouse. The data model is designed to be used in conjunction with reference data: standard instances that represent information common to a number of users, process plants, or both. The support for a specific life-cycle activity depends on the use of appropriate reference data in conjunction with the data model.

ISO 15926 is organized as a number of parts, each published separately. This part of ISO 15926 specifies a conceptual data model for computer representation of technical information about process plants.

The organization of this part of ISO 15926 is as follows:

- clause 1 specifies the scope and field of application of this part of ISO 15926;
- clause 2 identifies additional standards that, through references in this part of ISO 15926, constitute provisions of this part of ISO 15926;
- clause 3 defines terms used in this part of ISO 15926;
- clause 4 provides an overview of the fundamental concepts and assumptions that form the basis for the data model;
- clause 5 specifies the data model using the EXPRESS language (ISO 10303-11) and contains the EXPRESS-G diagrams that illustrate the structure of the model.

Readers of this part of ISO 15926 require knowledge of the information used by the designers, constructors and operators of process plants; an understanding of conceptual data models and the EPISTLE[1] data modelling principles [5]; and knowledge of the EXPRESS language.

In this International Standard the same English language words may be used to refer to a real world thing or to an EXPRESS data type that represents the real world thing. These uses are distinguished by typographic convention. If a word or phrase occurs in the same typeface as the surrounding narrative text, the word or phrase refers to the real world thing. If the word or phrase occurs in bold typeface, it refers to the EXPRESS type. Names of EXPRESS schemas also occur in bold typeface.

In the definitions of entity data types declared in this document, the wording “A <entity data type name> is ...” is used as a synonymous phrase for “A member of the class represented by the <entity data type name> entity data type is ...”.

[1] EPISTLE is the European Process Industry STEP Technical Liaison Executive.

Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities —

Part 2: Data model

1 Scope

This part of ISO 15926 specifies a conceptual data model for computer representation of technical information about process plants.

The following are within the scope of this part of ISO 15926:

- specification of requirements to produce, process, and transport process materials;
- specification of functions required to produce and process the required materials, including the following:
  - hydrocarbon process and conditioning systems,
  - injected gas and water conditioning and injection systems,
  - oil and gas product transport systems,
  - safety and control systems,
  - electricity generation and supply systems,
  - steam generation and supply systems,
  - structures,
  - buildings and accommodation;
- specification and selection of materials and equipment to provide the required production and processing functions, including information about market available materials and equipment;
- installation and commissioning of plant equipment;
- production and process operations, including process conditions and consumption, yields and quality of process material;
- maintenance and replacement of equipment.

The following are outside the scope of this part of ISO 15926:

- construction of buildings, production facilities and equipment.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601:2000, *Data elements and interchange formats — Information interchange — Representation of dates and times*

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

ISO 10303-1, *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 15926-1: –[2], *Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 1: Overview and fundamental principles*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply; those taken or adapted from other standards are repeated below for convenience.

NOTE Definitions copied verbatim from other standards are followed by a reference to the standard in brackets, such as "[ISO 10303-1]". In these cases the definition in the referenced document is normative; its repetition here is informative and in the case of any discrepancy the definition in the referenced document has precedence.

3.1.1

class
category or division of things based on one or more criterion for inclusion and exclusion

NOTE 1 A class need not have any known members (things that satisfy its criteria for membership).

NOTE 2 Because of the spatio-temporal paradigm is used to define individuals in this part of ISO 15926, all classes are non-well-founded sets. These are explained in D.2.4

[ISO 15926-1]

[2] To be published.

3.1.2

conceptual data model
data model in the three schema architecture defined by ISO/TR 9007 [2], in which the structure of data is represented in a form independent of any physical storage or external presentation format

[ISO 15926-1]

3.1.3

data
representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers

[ISO 10303-1]

3.1.4

data store
computer system that allows data to be stored for future reference

[ISO 15926-1]

3.1.5

data warehouse
data store in which related data are merged to provide an integrated set of data containing no duplication or redundancy of information, and which supports many different application viewpoints

[ISO 15926-1]

3.1.6

individual
thing that exists in space and time

NOTE 1 In this context, existence could be within the world we live in, or some "possible" world that can be imagined. This therefore includes actual, hypothetical, planned, expected, or required individuals.

EXAMPLE A pump with serial number ABC123, Battersea Power Station, Sir Joseph Whitworth, and the Starship “Enterprise” are examples of individuals.

NOTE 2 See 4.6.2 and 4.7 for a detailed discussion of the concept of individual.

[ISO 15926-1]

3.1.7

information
facts, concepts, or instructions

[ISO 10303-1]

3.1.8

process plant life-cycle data
data that represents, in computer processable form, information about one or more process plants in or throughout any phase or phases of a process plant's life-cycle, including design, engineering, construction, operation, maintenance, decommissioning and demolition

[ISO 15926-1]

3.1.9

reference data
process plant life-cycle data that represents information about classes or individuals which are common to many process plants or of interest to many users

[ISO 15926-1]

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply.

UTC	Coordinated Universal Time

4 Fundamental concepts and assumptions

4.1 Conceptual data model

The data model specified in clause 5 is a conceptual data model. Figure 1 shows its relationship to internal and external models (see ISO 15926-1:5.2).

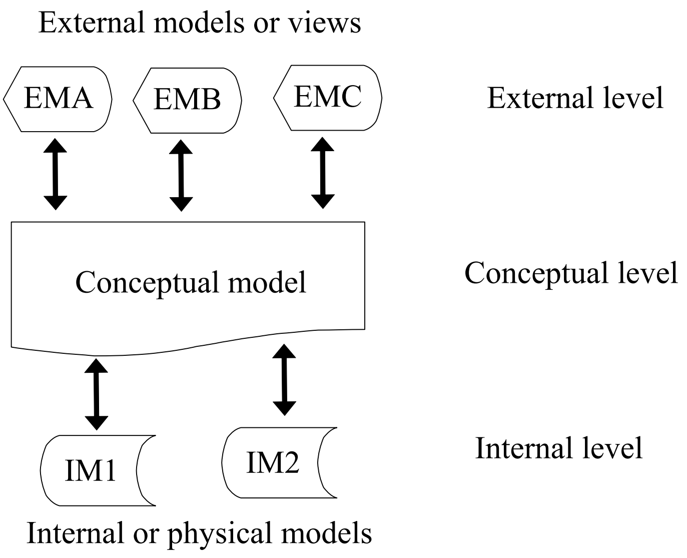


Figure 1 — Three level architecture

NOTE 1 The term conceptual data model is defined in ISO 15926-1 and is based on the three-schema architecture as described in ISO/TR 9007 [2].

To enable integration of life-cycle process plant information the model excludes all information constraints that are appropriate only to particular applications within the scope.

NOTE 2 Data integration means combining information derived from several independent sources into one coherent set of data that represents what is known. Because the independent sources often have overlapping scopes, combining their data requires the common things to be recognized, duplicate information to be removed, and new information represented. To succeed in the role of integration, the data model must have a context that can include all the possible data that might be wanted or required.

4.2 Data model design

This data model is designed in accordance with data modelling principles [5] developed by the European Process Industries STEP Technical Liaison Executive (EPISTLE). These principles control the use of entity data types, attributes and relationships when defining a conceptual data model.

The effects of these principles are as follows.

- The model entity data types are part of a universal subtype/supertype hierarchy of entity data types.

- Entity data types are generic, representing and being named after the persistent nature of their members.

- Attribute information is usually expressed by references to entity data types.

- Attributes that are representations of numbers, text characters and binary patterns, are defined as EXPRESS simple types.

- Relationships and activities are represented by entity data types.

4.3 System identifiers

The data model includes an artificial, system defined, system identifier for each entity instance - **thing.id**. Each such identifier is a member of the EXPRESS string data type and is constrained to be unique within the set of identifiers managed by a given system. System identifiers are mandatory and shall remain consistent for the life of every entity instance managed by a given system.

EXAMPLE A database system that is used to manage information about the design and engineering of an offshore production platform provides the scope of uniqueness for system identifiers within that database.

Each system identifier shall be interpreted as the system reference to the **thing** it is identifying.

Other identifiers, external to the system, used for a particular **thing** are recorded using the **class_of_identification** entity data type.

4.4 Record management information

The data model makes provision for holding limited information about the computer records used to represent process plant information. The record management data is specified as attribute values that may be given for each entity instance. The scope of that information is as follows:

- for records that originate in another system, the date and time when this copy of the record was created in the current system;

- the date and time on which this record was first created in its originating system;

- the person, organisation, or system that first created this record in the originating system;
- the date and time that this record was logically deleted;
- for logically deleted records, the reason why the record was logically deleted, where logical deletion means that whilst the record is still available in the system as a matter of historical record, it is now recognised as always being an invalid statement.

The definitions of the record management attributes are given in 5.2.1.2.

4.5 Documentation conventions

4.5.1 Entity and attribute definitions

Clause 4.6 introduces the concepts of the data model. The model is formally specified in clause 5. The introduction to 4.6 describes the model concepts and gives examples of their application. The descriptions in 4.6 complement the definitions and examples of clause 5, with clause 5 being the definitive description of the model.

4.5.2 Diagrams

This clause makes extensive use of diagrams to support the description of the fundamental concepts and assumptions.

Three types of diagrams are used:

- space-time maps;
- model diagrams, based on a subset of the EXPRESS G conventions;
- instance diagrams.

4.5.2.1 Space-time maps

Space-time maps are used to illustrate the use of space-time extensions to model tangible concrete objects that exist in space and time. Figure 2 is an example of a space-time map.

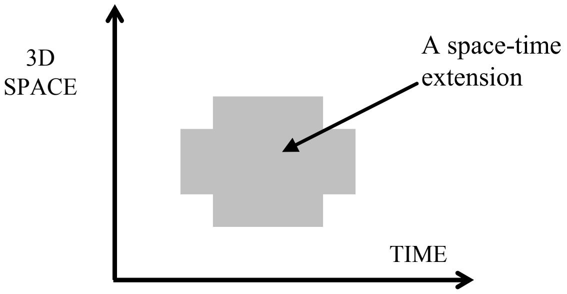


Figure 2 — Space-time map

A space-time map diagram consists of two perpendicular axes, the vertical axis for 3D space and the horizontal axis for time. Space-time extensions are plotted as bounded or shaded areas within these axes. The boundary to the left hand side of the extension indicates the start of the extension, and the boundary to the right hand side of the extension indicates the end of the extension. The change in boundaries at the top and bottom indicate changes in spatial extent.

4.5.2.2 Model diagrams

Model diagrams are used to illustrate portions of the model to support its explanation and understanding. Model diagrams are restricted to show: EXPRESS entity types; EXPRESS subtyping; and EXPRESS relationships and use the appropriate EXPRESS G symbols.

EXAMPLE Figure 3 is an example of a model diagram.

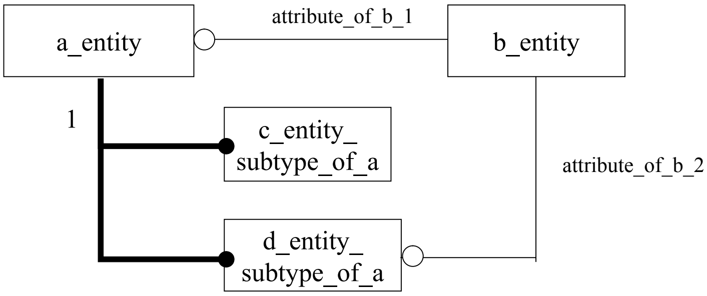
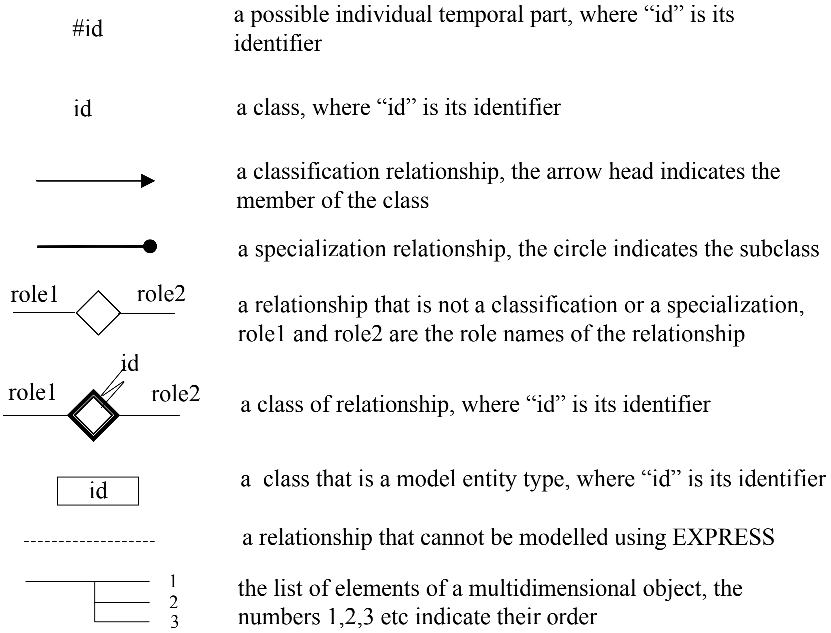


Figure 3 — Model diagram

A full set of EXPRESS G diagrams for the model is given in clause 5.

4.5.2.3 Instance diagrams

The meaning of the model entity types is illustrated using instance diagrams. The symbols used to construct instance diagrams are described in Figure 4.



#id	a possible individual temporal part, where “id” is its identifier
id	a class, where “id” is its identifier
⟶	a classification relationship, the arrow head indicates the member of the class
—●	a specialization relationship, the circle indicates the subclass
role1 ◇ role2	a relationship that is not a classification or a specialization, role1 and role2 are the role names of the relationship
role1 ◈ role2 id	a class of relationship, where “id” is its identifier
□ id	a class that is a model entity type, where “id” is its identifier
------------------	a relationship that cannot be modelled using EXPRESS
1 2 3	the list of elements of a multidimensional object, the numbers 1,2,3 etc indicate their order

Figure 4 — Instance diagram notation

Figure 5 is an example instance diagram. It shows an object identified as #1234 that is a member of **thing**, and a member of pump. ‘Pump’; is a member of **class**. Both **thing** and **class** are entity types of the model.

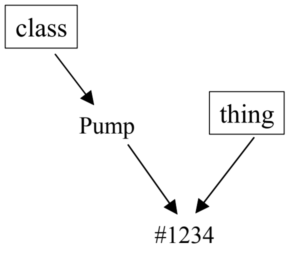


Figure 5 — Example diagram notation

NOTE 1 The symbols in Figure 4 enable examples of **thing**, entity data type, **classification**, **specialization**, **relationship**, and **class_of_relationship** to be represented in this part of ISO 15926.

NOTE 2 In this part of ISO 15926, the model entity data types defined in clause 5 are considered to be classes. Where such classes are used in illustrations, they are shown as a rectangular box.

NOTE 3 In this clause, the instance diagrams are considered complete when the objects shown are entity data types or are direct or indirect members of entity data types. The relationships between the model entity data types that are defined in the model are not shown.

NOTE 4 The instance diagrams in this clause are not intended to show how the model would be instantiated in practice. In particular, membership of the most specific model entity types may not be shown and some entity type memberships may be omitted altogether.

4.6 Data model concepts

4.6.1 Thing

The data model consists of a universal subtype/supertype entity type hierarchy.

There is one root supertype named **thing** that is the class of everything.

things are subdivided into

— either, **possible_individual**s;

— or, **abstract_object**s.

The topmost subdivisions of this hierarchy are shown in Figure 6 (see also 5.2.1 and Figure 177).

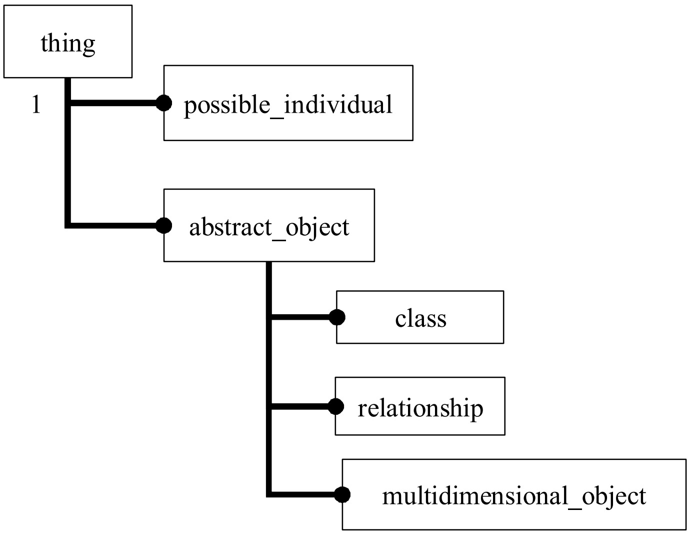


Figure 6 — Part of the model subtype/supertype hierarchy

4.6.2 Possible individual

A **possible_individual** is a **thing** that exists in space and time (see Figure 6 and 5.2.6). The identity of a **possible_individual** is its space-time extension. No two individuals have exactly the same space-time extension. Everyday physical things, often referred to as concrete objects, are **possible_individual**s.

EXAMPLE The pump known by the serial number #1234 is a **possible_individual**.

In contrast an **abstract_object** is a **thing** that does not exist in space-time. **class**, **relationship** and **multidimensional_object** are kinds of **abstract_object**.

4.6.3 Class

A **class** is an understanding of the nature of **thing**s; that divides **thing**s into **thing**s that are members of the **class** and **thing**s that are not, according to one or more criteria (see 5.2.2 and Figure 178). The identity of a **class** is its membership. No two **class**es have the same membership.

EXAMPLE 1 The concept known as pump is a **class**.

Classes are universal with no space-time extent. However, **class**es may involve time and space as criterion.

EXAMPLE 2 ‘Sales in June’ is a **class**.

4.6.4 Relationship

A **relationship** is something that one **thing** has to do with another (see 5.2.11 and Figure 187). In this part of ISO 15926, a **relationship** is defined as the classification of an ordered pair. The pair is repeated to record another **relationship**. No two **relationship**s of the same classification have the same pair in the same order. The order enables roles to be assigned to the related **thing**s.

EXAMPLE 1 The pair consisting of ‘pump’ and ‘#1234’ in the order where ‘pump’ acts as the **class** and ‘#1234’ acts as the member is a **classification** relationship.

This part of ISO 15926 defines some explicit subtypes of **relationship**, covering some commonly used relationships in the process industries.

EXAMPLE 2 Explicit subtypes of **relationship** include **classification**, **specialization**, **lifecycle_stage** and **approval**.

4.6.5 Multidimensional object

A **multidimensional_object** is an ordered list of **thing**s (see 5.2.4 and Figure 180). The **thing**s in the list can be **possible_individual**s, **class**es, **relationship**s or other **multidimensional_object**s.

EXAMPLE The list [2.0, 4.0, 5.7] is a **multidimensional_object**.

The order of the elements of a **multidimensional_object** is defined using the EXPRESS LIST aggregate type.

4.7 Possible individual

A **possible_individual** is a **thing** that could exist in space and time, often referred to as a concrete object (see 5.2.6 and Figure 182).

Possible_individual includes **thing**s that:

- actually exist, or have actually existed in the past;
- possibly have existed in the past, and may possibly exist in the future;
- are hypothetical having no existence in the past or future.

In this International Standard, a **possible_individual** corresponds to a particular extension in space and time. Things that have coincident space-time extensions are considered to be the same thing.

The general nature of a **possible_individual** is illustrated by the space-time map for the **possible_individual** '#1234' shown in Figure 7.

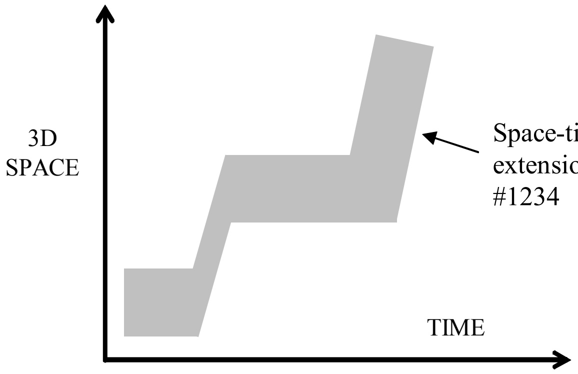


Figure 7 — Possible individual as a space-time extension

The instance diagram that represents the particular space-time extension in Figure 7 is shown in Figure 8.

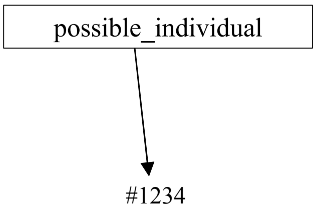


Figure 8 — Instance diagram for possible individual #1234

NOTE In this International Standard, “possible individual” and “individual” are terms used for space-time extension.

4.7.1 Composition of possible individual

A **possible_individual** may be part of other **possible_individual**s. Composition, or whole-part behaviour, distinguishes **possible_individual**s from **class**es. The concept of whole-part is represented by **composition_of_individual,** a subtype of **relationship** (see 5.2.6.5 and Figure 182). This is shown in Figure 9.

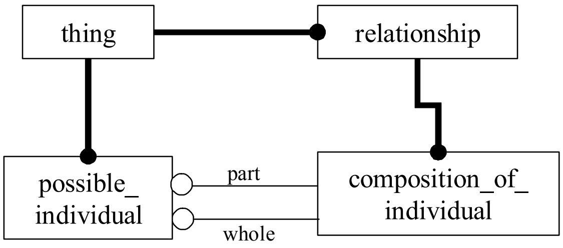


Figure 9 — Composition of individual relationships

EXAMPLE Consider the impeller of a centrifugal pump. The time in the life of the impeller that it is installed in the pump is a space-time extension that is part of both the pump and the impeller. This is illustrated in the space-time map shown in Figure 10. Extensions #1234 and #5678 represent the impeller and the pump respectively. Their intersection #9012 is where the impeller part is also part of the pump.

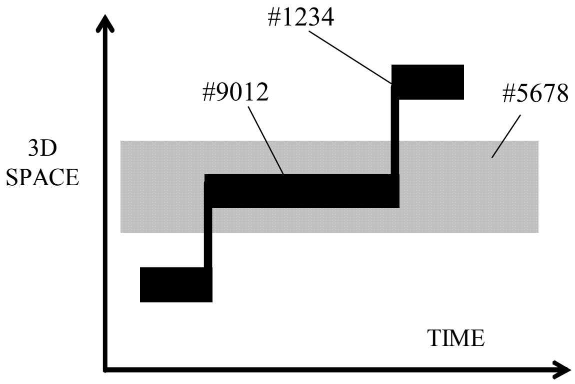


Figure 10 — Intersecting space-time extensions

The use of the model to represent the three space-time extensions is shown in Figure 11. The **possible_individual** #9012 is a part of both the **possible_individual**s #1234 and #5678.

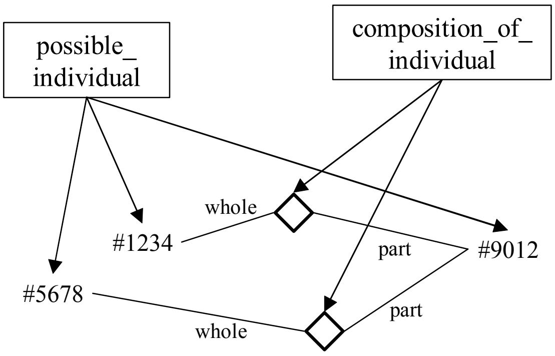


Figure 11 — Instance diagram for composition of individual

4.7.2 Temporal part of individual

In this part of ISO 15926, **possible_individual**s that correspond to the entire spatial extent for a period of time of another **possible_individual** are referred to as temporal parts (see 5.2.6.14 and Figure 182). The nature of a temporal part is illustrated in Figure 12.

NOTE State and substate are commonly used terms that are equivalent to temporal part.

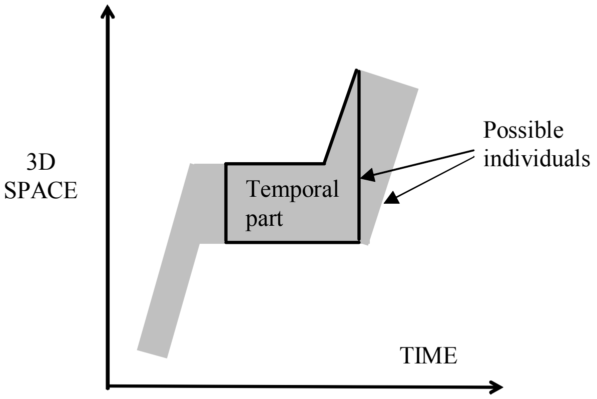


Figure 12 — Temporal part

The entity data type used to represent temporal parts is defined as a subtype of **composition_of_individual** as shown in Figure 13.

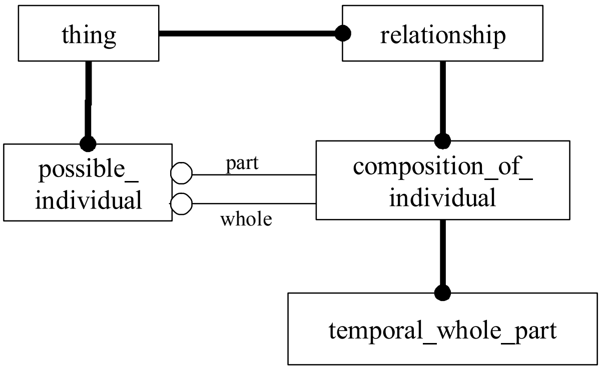


Figure 13 — Temporal whole part relationships

EXAMPLE In Figure 14, #9012 is a temporal part of the impeller #1234. The **composition_of_individual** relationship can be specialised to be **temporal_whole_part**.

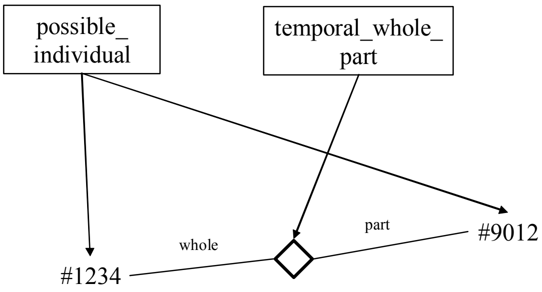


Figure 14 — Temporal part #9012 of #1234

With the exception of the concepts of wholeness and the nature of its temporal extent, all the characteristics of a **possible_individual** apply to all its temporal parts. However, the characteristics of a temporal part do not necessarily apply to the whole **possible_individual**.

4.7.3 Connection of individual

possible_individuals may be connected for their lifetime such that they can interact with each other (see 5.2.21 and Figure 197). The connection may be direct, that is they share a common space boundary; or it may be indirect via other **possible_individual**s. In the second case the intervening individuals will have a chain of direct connections that may not be recorded.

The space-time nature of a **direct_connection** is shown in Figure 15. The diagram shows two individuals that are in contact for a part of their lifetime. The temporal parts 'A' and 'B' of the two individuals have a common boundary in space for the duration of the connection. The boundary is shown as stationary with respect to time, though this not necessarily the case. 'A' and 'B' may also be considered to be parts of a third individual, 'W', not shown. The connection of 'A' and 'B' does not imply they are parts of a common whole, 'W'. This must be separately stated using **composition_of_individual** relationships. Similarly, being two parts of the same whole does not imply connection of the two parts.

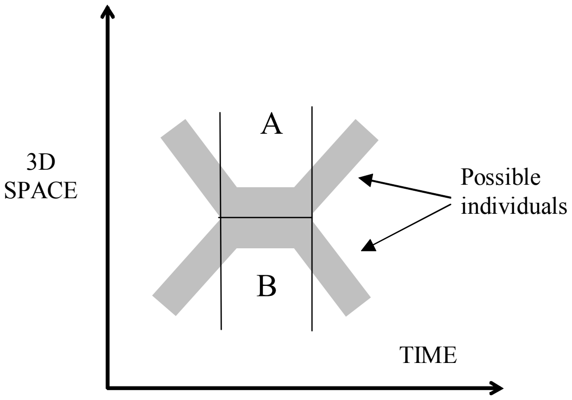


Figure 15 — Connected space-time extensions

The elements of the connection model are shown in Figure 16. **connection_of_individual** is a **relationship** with subtypes of **direct_connection** and **indirect_connection**. No order or direction is implied by the attribute names **side_1** and **side_2**. The names serve only to distinguish the individuals involved in the connection.

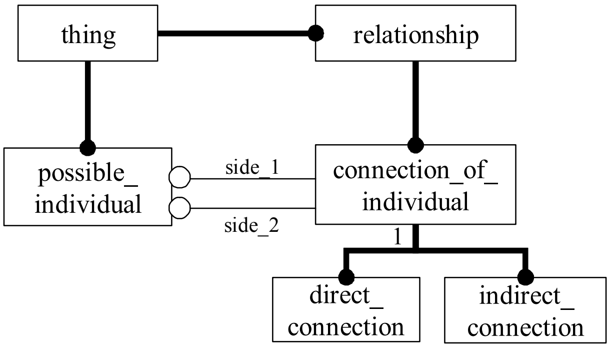


Figure 16 — Connection of individual

EXAMPLE 1 Figure 17 shows the connection between a particular engine shaft and a particular seal. The connection involves only temporal parts of the shaft and seal.

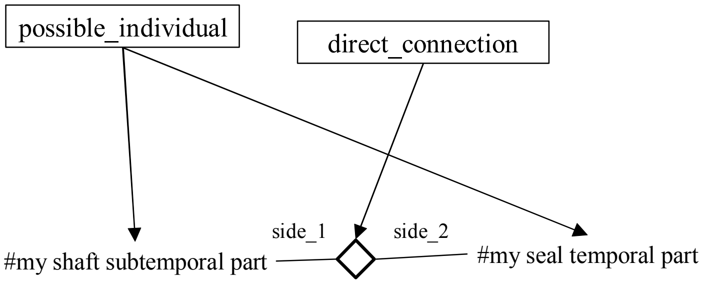


Figure 17 — Shaft seal direct connection

An **indirect_connection** is made via other individuals. Figure 18 shows the relationship **individual_used_in_connection**. This enables the other individuals involved in an **indirect_connection** to be recorded.

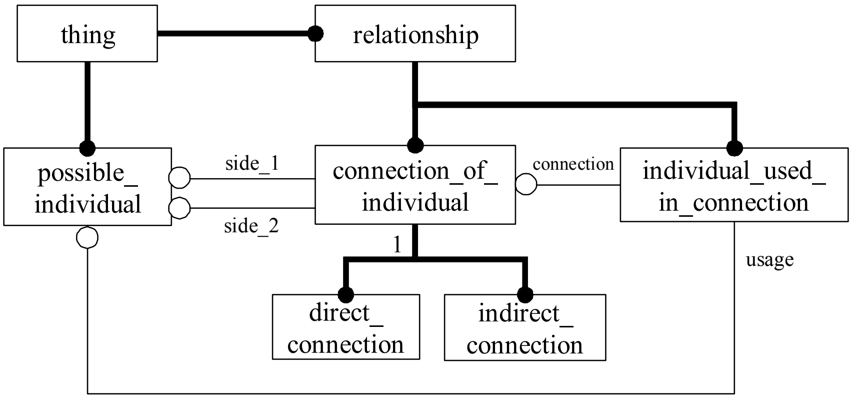


Figure 18 — Individual used in connection

EXAMPLE 2 Figure 19 shows an **indirect_connection** between the engine shaft and crankcase, where a bearing and seal is used to make the connection.

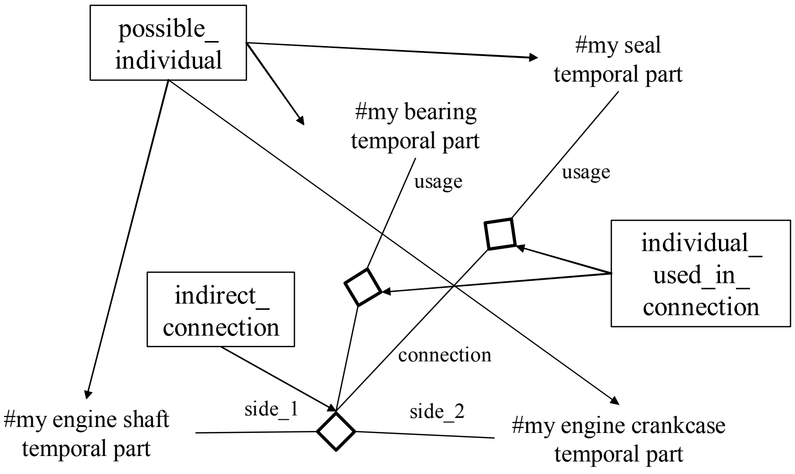

possible_ individual
#my seal temporal part
#my bearing temporal part
usage
usage
individual_ used_in_ connection
indirect_ connection
connection
side_1
side_2
#my engine shaft temporal part
#my engine crankcase temporal part

Figure 19 — Shaft crankcase indirect connection

4.7.4 Temporal sequence of individual

Space-time extensions may be ordered with respect to time, the entirety of one following the entirety of another. This is known as **temporal_sequence** in this part of ISO 15926 (see 5.2.22.6 and Figure 198) The space-time nature of **temporal_sequence** is shown in Figure 20. The figure shows individuals ‘A’ and ‘B’ which are sequenced in the time dimension where all of ‘B’ comes after all of ‘A’.

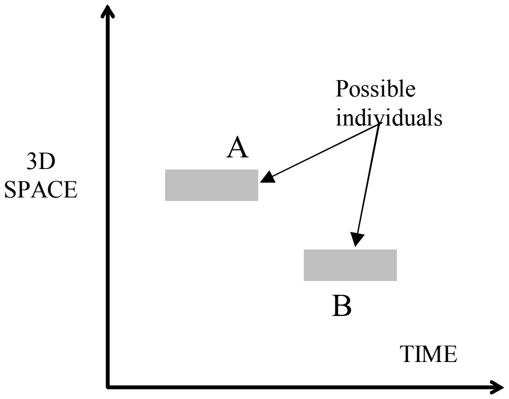


Figure 20 — Sequence of space-time extensions

The elements of the sequence model are shown in Figure 21. **temporal_sequence** is a type of **relationship** between a predecessor and successor **possible_individual** defining a sequence in time.

Sequence only records the case of what is. Rules concerning sequence, in the sense that the nature of a type of thing is that the members are in a sequence with another, are expressed using **class_of_temporal_sequence** (see section 4.8.4.7 and 5.2.22.3).

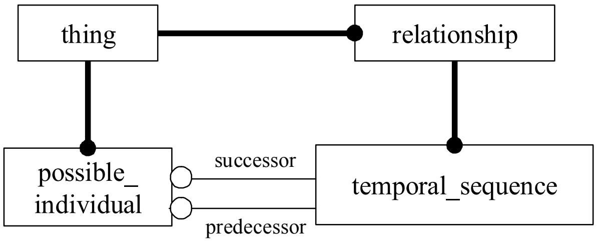


Figure 21 — Temporal sequence

EXAMPLE Figure 22 below shows the **possible_individual** known as James Watt came after the Battle of Hastings.

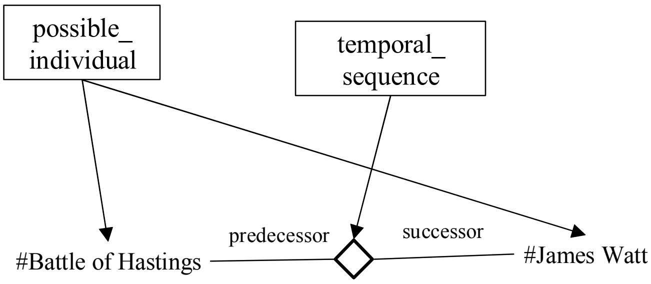


Figure 22 — James Watt and the Battle of Hastings

4.7.5 Subtypes of individual

The direct subtypes of **possible_individual** defined in the model are shown in Figure 23 (see also Figure 182).

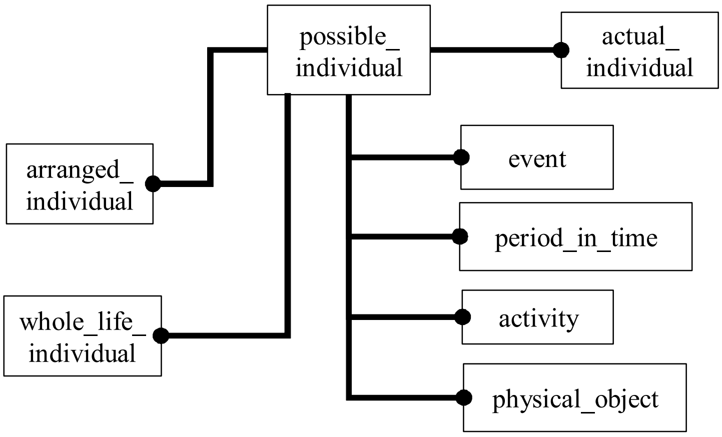


Figure 23 — Subtypes of possible_individual

4.7.6 Actual individual

actual_individuals are **possible_individual**s that actually exist in the real world (see 5.2.6.1). **thing**s that we may have planned in the past, or plan or expect in the future that never happen remain as just **possible_individual**s. Only those **possible_individual**s that come to pass are also **actual_individual**s.

EXAMPLE The 25 gallon per minute pump that was required for the Omega production system from 21 March 1998 is a **possible_individual**. The pump manufactured in July of 1998 that has a capacity of 28 gallon per minute is an **actual_individual**. The extensions for the required 25 gpm pump and the actual 28 gpm pump are shown in Figure 24. Because they have different space-time boundaries, they are different extensions.

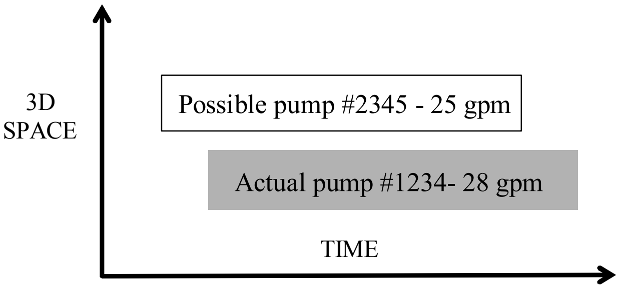


Figure 24 — Possible and actual individuals

Figure 25 shows the instance diagram for these extensions. Two **possible_individual**s are shown #2345 and #1234, each corresponding to a distinct extension. #1234 is a member of both the classes **actual_individual** and **possible_individual**.

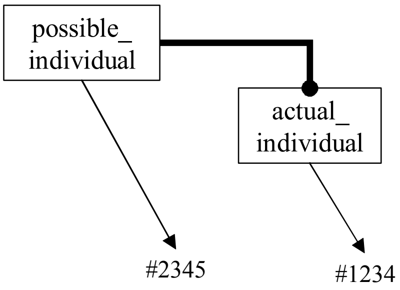


Figure 25 — Instance diagram of possible and actual individuals

4.7.7 Lifecycle stage of individual

Organisations and people often refer to **possible_individual**s using the lifecycle concepts such as proposed, planned or required. In this part of ISO 15926, **lifecycle_stage** is modelled as a relationship between two **possible_individual**s (see 5.2.23.4, 5.2.23.5 and Figure 199). This is shown in Figure 26.

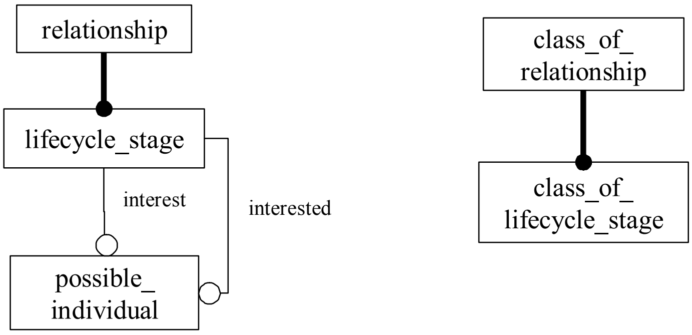


Figure 26 — Lifecycle relationship

There are no constraints on the relationship members of a **class_of_lifecycle_stage**.

EXAMPLE Figure 27 shows the 25 gpm pump #2345 being required by the XYZ Company during the period 6 January to 27 September 2002. The same **possible_individual** could also be regarded as being required or planned by another organisation.

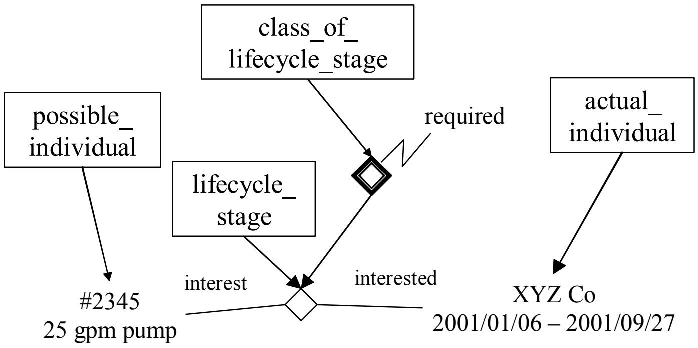


Figure 27 — The pump required by XYZ Co

4.7.8 Whole life individual

A **whole_life_individual** is a space-time extension that is not a temporal part of any other individual that is of the same class (see 5.2.6.15 and Figure 182). They are individuals whose identities are considered to be independent of other individuals.

EXAMPLE 1 The actual impeller with serial number #1234 is a **whole_life_individual**. It is not a temporal part of another impeller.

EXAMPLE 2 Consider a plastic blank, a cup formed from the blank, and some crushed plastic formed from the cup. They are temporal parts of the individual that is the plastic piece whose molecules are common to the blank, cup and crushed waste. Because the cup is not of the same class as the plastic piece, it is considered to be a **whole_life_individual**. For similar reasons the plastic blank and the crushed plastic are also **whole_life_individuals**. This is illustrated by Figure 28.

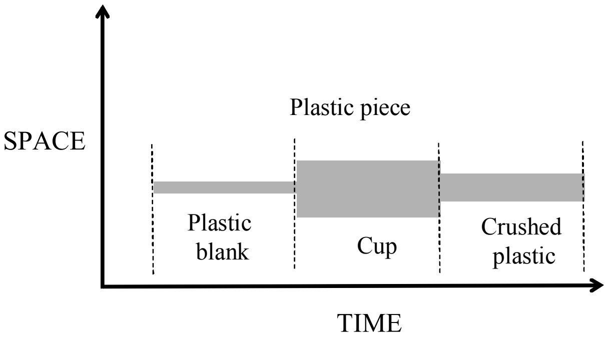


Figure 28 — Space-time map of a piece of plastic

EXAMPLE 3 Figure 29 shows the model instances necessary to represent the cup #X93 and the plastic piece #3A as **whole_life_individuals**, where the cup plays the role of part and the plastic piece the role of whole in the **temporal_whole_part** relationship.

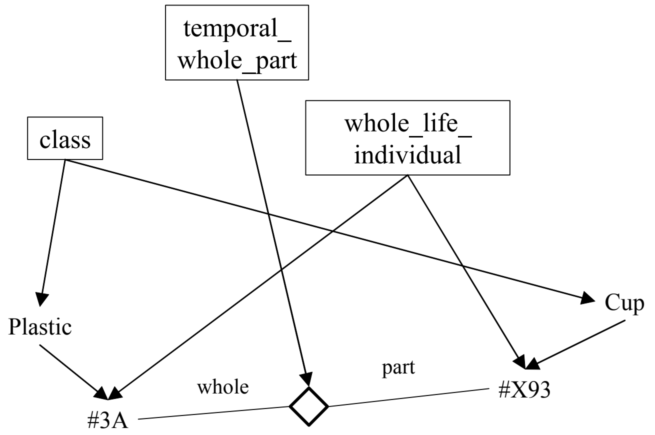


Figure 29 — Plastic piece and cup as whole life individuals

4.7.9 Arranged individual

possible_individuals are composed of other **possible_individual**s, referred to as parts. Those where the parts have a particular organization, or arrangement, are **arranged_individual**s (see 5.2.6.2 and Figure 182). The properties, characteristics and behaviours of an **arranged_individual** are different from those of its individual parts.

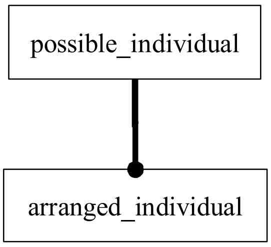


Figure 30 — Arranged individual

EXAMPLE 1 A particular pump with a manufacturer’s serial number is a **arranged_individual**. The pumping capablity is its behaviour that is not present in any of its individual parts.

EXAMPLE 2 The stock of spare pump impellers is not an **arranged_individual**. They have no intended aggregate behaviour.

4.7.9.1 Arrangement of individual

arrangement_of_individual is defined as a subtype of **composition_of_individual**, constrained to refer to **arranged_individual**s as the **whole** (see Figure 31). An **arrangement_of_individual** indicates that the **part** is arranged with respect to other parts of the **whole** (see 5.2.6.3 and Figure 182).

EXAMPLE 1 Several aircraft flying in formation is an **arranged_individual**. **arrangement_of_individual** relationships indicate the temporal parts of each aircraft that are part of the formation. When the aircraft are on the ground, they are not part of the flying formation, hence the formation is composed of temporal parts of the aircraft.

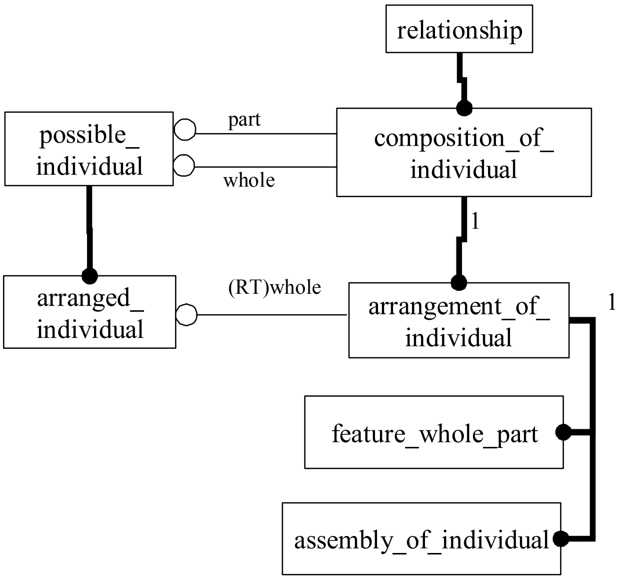


Figure 31 — Arrangement of individual

Two subtypes of **arrangement_of_individual** are defined to distinguish piece part assembly and feature parts of **arranged_individual**s.

assembly_of_individual relationships indicate that the parts of the whole are directly connected temporal parts of the components (see 5.2.6.4). The nature of the implied connection is that the part can be reasonably joined and separated from the whole particularly by mechanical means or by welding, gluing, and other forms of adhesion. This allows the parts to be replaced during the lifetime of the **arranged_individual** and for the parts to be parts of other **arranged_individual**s.

EXAMPLE 2 Figure 32 shows the **assembly_of_individual** relationships between two temporal parts of the impeller #I27C, designated as A and B, and the pumps, shown as #2345 and #2346, of which they are a part. #2345 and #2346 are temporal parts of two other whole individuals that would also be classified as pumps but are not shown here.

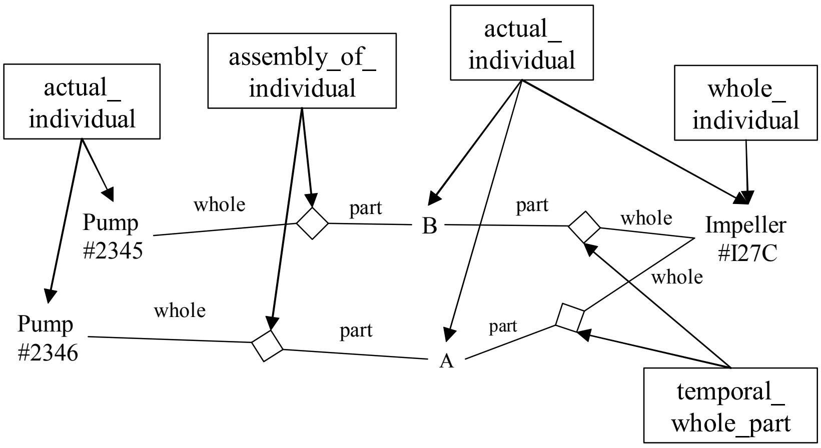


Figure 32 — Assembly of pump

In this part of ISO 15926, instances of **assembly_of_individual** are restricted to mesoscopic sized individuals, and therefore exclude composition of molecular and atomic sized individuals.

feature_whole_part relations apply to parts of **arranged_individual**s that are non separable from the whole (see 5.2.6.6). The feature is such that it has no identification other than as a part of the individual of which it is a feature. In terms of this part of ISO 15926, a feature part is either a **whole_life_individual** or a part of a **whole_life_individual** that is a feature part of another **arranged_individual**.

EXAMPLE 3 Figure 33 shows the data for a corroded surface section of a pipeline. The surface section is a **whole_life_individual** that is a feature of the pipeline. The corroded section has individual parts that reflect the spatial extent and severity of the corrosion. The temporal part inherits the condition that it is a feature of the pipe. The development of corrosion can be tracked through time by identifying an appropriate whole life object and making the temporal parts that may be observed or measured from time to time, temporal parts of the **whole_life_individual**.

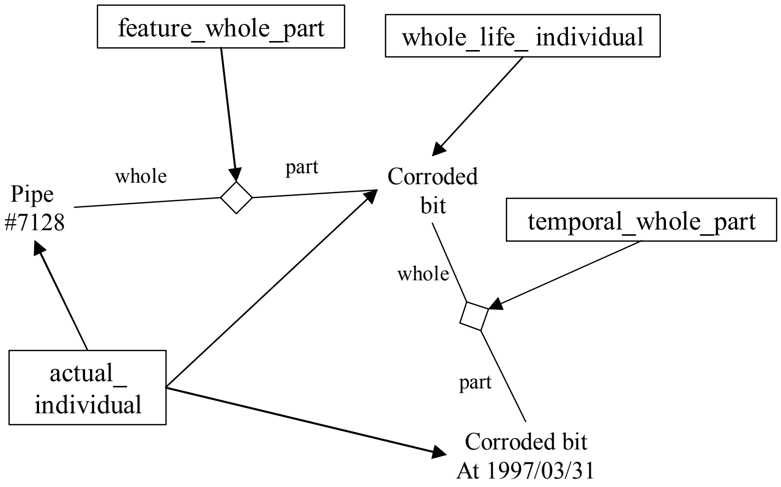


Figure 33 — Corrosion features

4.7.10 Event and point in time

In this part of ISO 15926, **event**s are defined as space-time extensions with zero time extension (see 5.2.9.5 and Figure 185). **event**s may be at one-time only, or may be continuous in time, or a combination of both. Figure 34 shows a space-time map for one-time and continuous **event**s. Both types satisfy the condition of zero time extension as every part has zero duration.

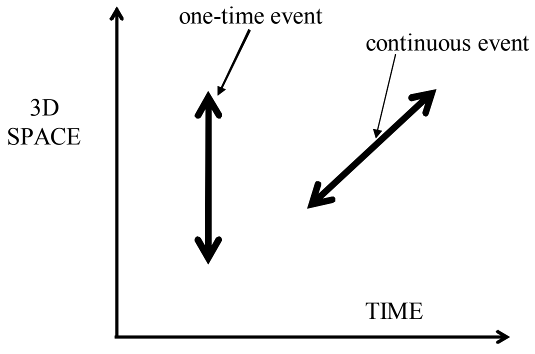


Figure 34 — Event space-time extensions

events mark the temporal boundaries of **possible_individual**s. Figure 35 shows the space-time map of an object that is moving, then at rest, and then moving. The one-time **event**s ‘A’ and ‘B’ delineate the stationary temporal part of the object. The continuous **event** ‘C’ marks the boundary of the leading edge of the moving object. **event** 'D' that has parts 'C' and 'B', is the beginning temporal boundary of the moving temporal part of the object.

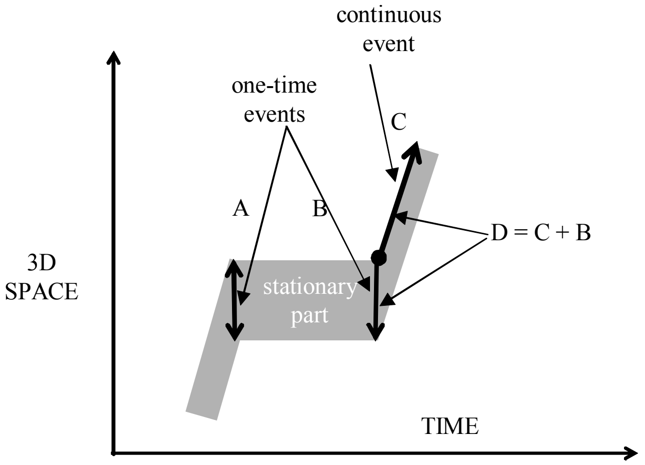


Figure 35 — Event boundary space-time map

The model of **event** is shown in Figure 36. **event** is a subtype of **possible_individual** and so can also be an **actual_individual**. An **event** is a temporal boundary of other space-time extensions. **temporal_bounding** is defined as a subtype of **composition_of_individual**, where the part is constrained to be an **event**. The subtypes **beginning** and **ending** indicate the **temporal_sequence** of the boundary relative to the bounded individual.

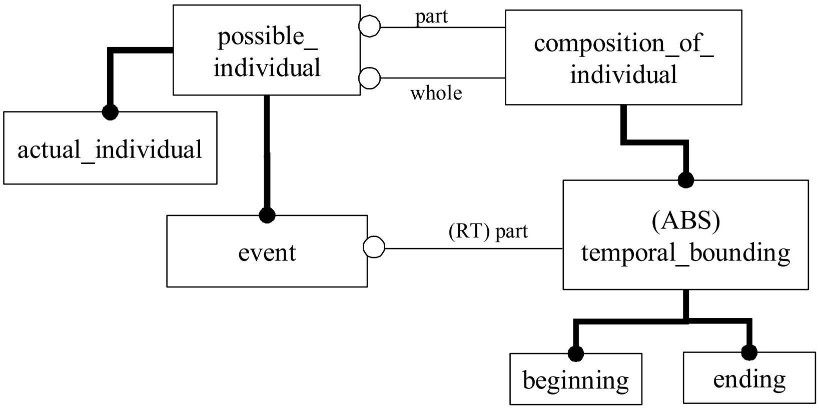


Figure 36 — Model diagram of event

EXAMPLE 1 A pipeline pig is brought to rest, parked for a period, before moving off again. The end of the pig's stationary state is a one time **event** and an **actual_individual**. The data for this is shown in Figure 37 below.

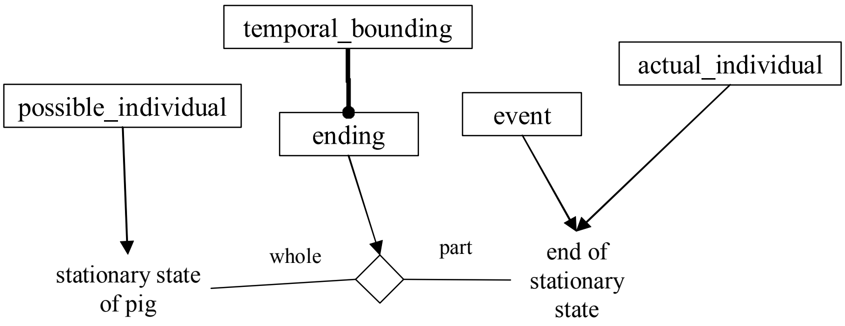


Figure 37 — Instance diagram of the ending of the stationary state

EXAMPLE 2 Figure 38 shows the space-time trajectory of the front edge of the pig for the period of motion. The start of the moving state is the end of the stationary state. Both are parts of the **beginning event** of the moving state.

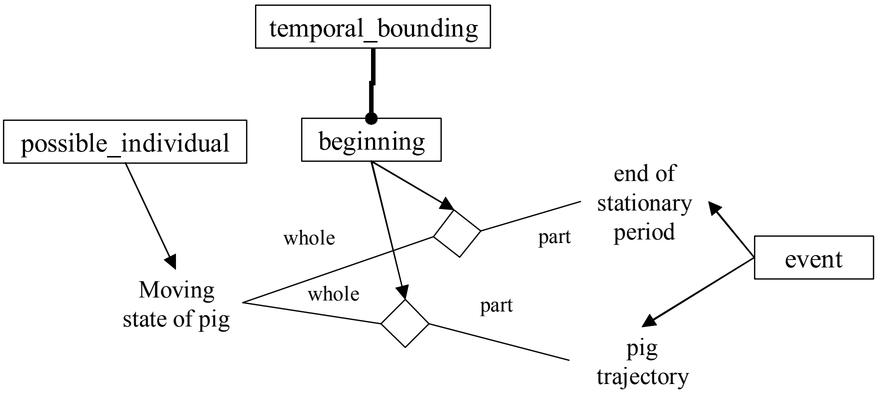


Figure 38 — Instance diagram of the pig space-time trajectory

Some one-time **event**s include the entire space extension; these are known as **point_in_time** (see 5.2.9.8 and Figure 185). One-time **event**s are always a part of a **point_in_time**. This is illustrated in Figure 39.

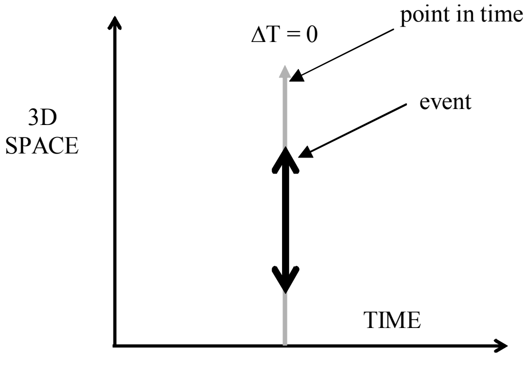


Figure 39 — Point in time extensions

The model of **event** and **point_in_time** is shown in Figure 40. As **event**s and **point_in_time** are **possible_individual**s, they may also be **actual_individual**s.

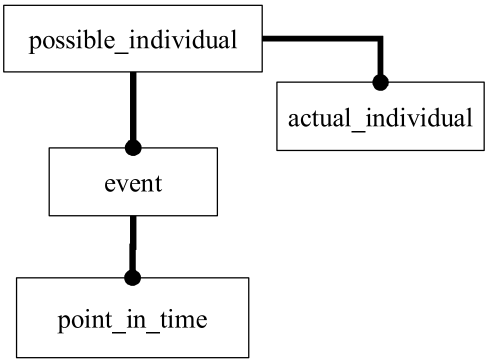


Figure 40 — Model diagram of Event

EXAMPLE 3 The time known as 10am 17 November 2002 UTC that has occured is a **point_in_time** and an **actual_individual** (see Figure 41).

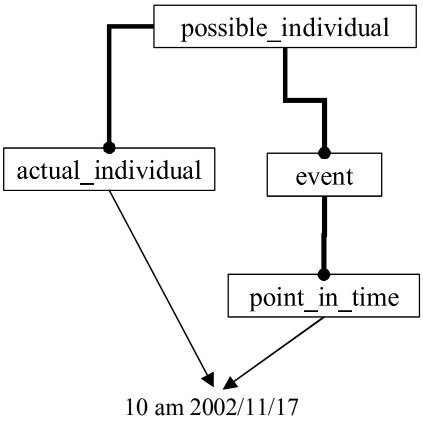


Figure 41 — Instance diagram of the actual point in time described as 10am 17 November 2002 UTC

events that are not points in time are spatial parts of a **point_in_time**, defining the time of the **event**. The **composition_of_individual** relationship is used to represent this.

EXAMPLE 4 Figure 42 shows the stationary state of the pipeline pig began its existence, at 10 am 2002/11/17 UTC.

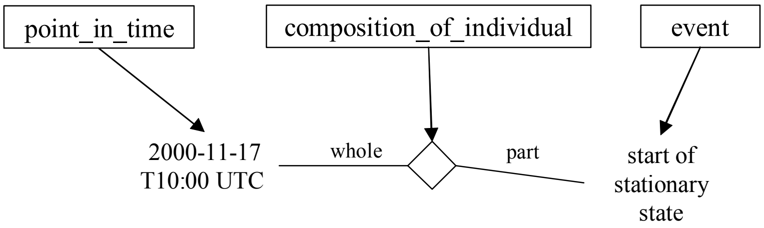


Figure 42 — Instance diagram of time of the stationary state

4.7.11 Period in time

A **period_in_time** is a **possible_individual** that is all space for part of time - a temporal part of the universe (see 5.2.6.9 and Figure 182). The space-time nature of **period_in_time** is illustrated in Figure 43.

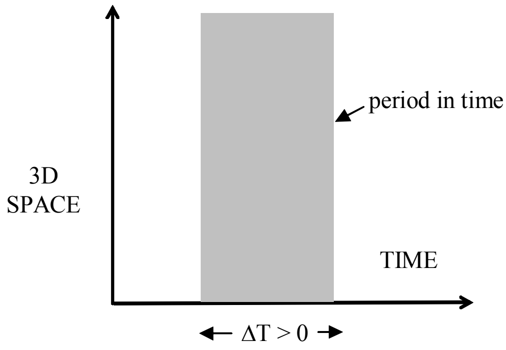


Figure 43 — Period in time space-time extension

The model for **period_in_time** is shown in Figure 44.

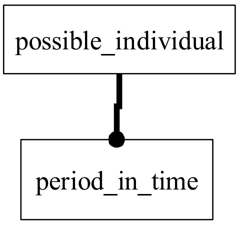


Figure 44 — Period in time entity type

Periods in time are bounded by a **beginning** and **ending** point in time as shown in Figure 45.

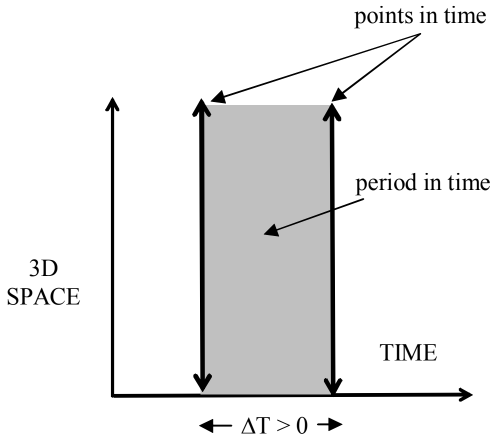


Figure 45 — Space-time map for a period of time and its bounding points in time

EXAMPLE In Figure 46, #EH26 is a **period_in_time**,with starting time 10:26 and ending time 11:09.

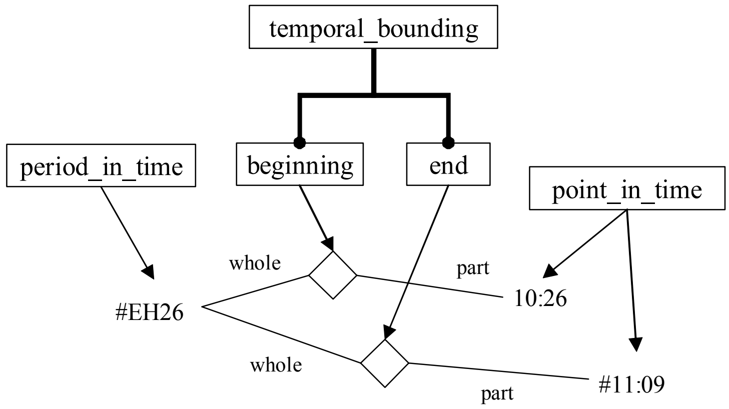


Figure 46 — Period in time 10:26 to 11:09

4.7.12 Physical object

In this part of ISO 15926, a **physical_object** is a distribution of matter and/or energy in time and space (see 5.2.6.10 and Figure 182). **physical_object**s are always a part of space and usually, though not necessarily, have a non-zero temporal extension.

physical_object and **activity** are not mutually exclusive. A **possible_individual** can be both.

EXAMPLE 'Radioactive materials', 'living organisms' and 'fire' are examples of **physical_object** and **activity**.

Four kinds of **physical_object** are recognized, discriminated on their basis of continuity. They are defined as subtypes of **physical_object** as shown in Figure 47:

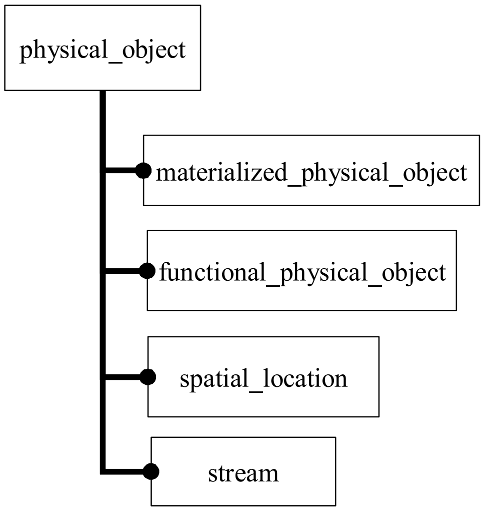


Figure 47 — Types of physical object

4.7.13 Materialised physical object

materialized_physical_objects are **physical_object**s that consist of the same or slowly changing material or energy during their existence (see 5.2.6.8 and Figure 182). This includes most of what are often described as concrete objects.

EXAMPLE A mechanical pump bearing the manufacturer's serial number is a **materialized_physical_object**. It is regarded as the same thing even if all of its components are replaced during its life. Figure 48. shows a space-time map for a pump initially comprising two parts A & B. After some time, component A is replaced by a new component C. Later B is replaced by D. Material continuity is achieved as some material persists across each event boundary.

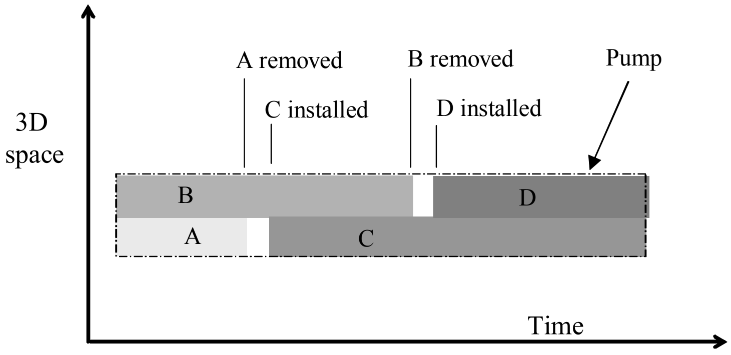


Figure 48 — Material continuity space-time map

The pump consists of a number of temporal parts each corresponding to the different component parts. This is illustrated in Figure 49.

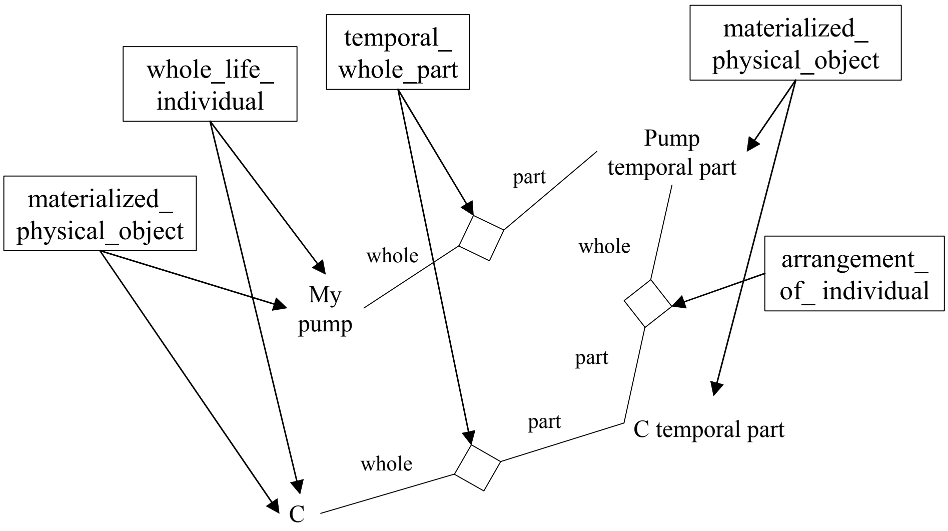


Figure 49 — Pump temporal parts

4.7.14 Functional physical object

functional_physical_objects are **physical_object**s based on continuity of intended function (see 5.2.6.7 and Figure 182). The material that makes up the object can be completely changed, provided the intended function of the individual remains the same.

temporal_whole_part relations are used to indicate which **materialized_physical_object** temporal parts are parts of a **functional_physical_object**.

EXAMPLE The pump known by the Tag 'P101', for which a succession of different pumping equipment items are installed, is a **functional_physical_object**. This is illustrated in Figure 50. pump 1 is installed and removed from Tag P101. Later pump 2 is installed and removed from Tag P101. There is no material continuity, but both pumps perform a similar function whilst installed. The instances required for pump 1 are shown in Figure 51.

The **possible_individual** that is a temporal part of both the tag and of the pump is both a **materialized_physical_object** and **functional_physical_object**.

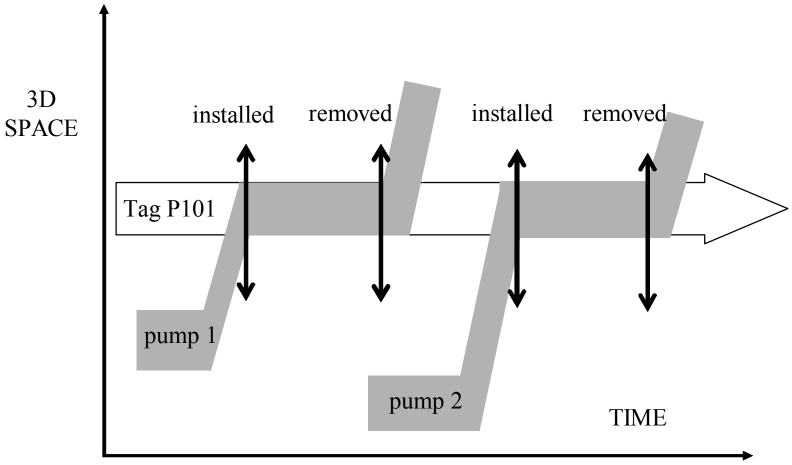


Figure 50 — Function physical object P101 space-time map

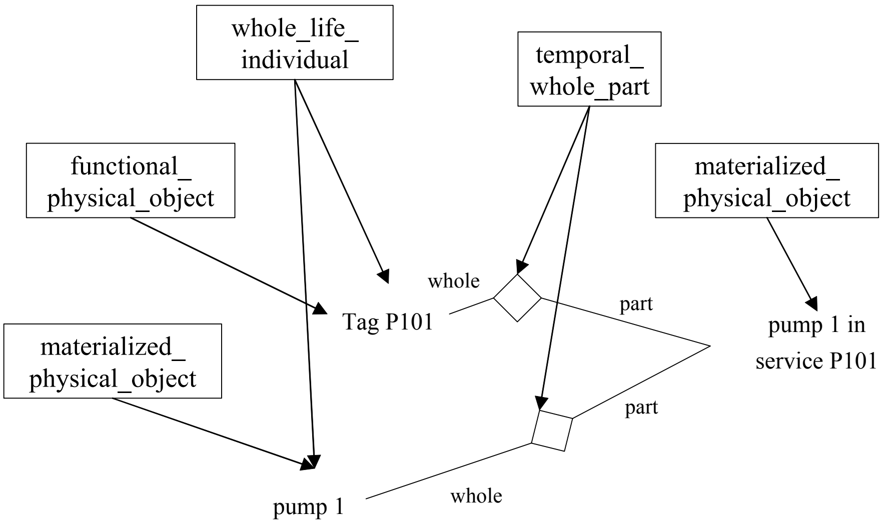


Figure 51 — Instance diagram for pump 1 installed as P101

4.7.15 Spatial location

spatial_locations are **physical_object**s where continuity of relative position is the basis of identity (see 5.2.6.12 and Figure 182).

EXAMPLE An offshore license area is a **spatial_location**.

4.7.16 Stream

streams are **physical_object**s where continuity of flow path is the basis of identity (see 5.2.6.13 and Figure 182).

EXAMPLE The moving contents of a hose is a **stream** and a **materialized_physical_object**.

4.7.17 Activity

An **activity** is something happening or changing (see 5.2.9.1 and Figure 185). In this part of ISO 15926, activities are considered to be space-time extensions in which other individuals and **event**s participate. **participation** is defined as a type of **composition_of_individual** as shown in Figure 52 (see 5.2.9.7).

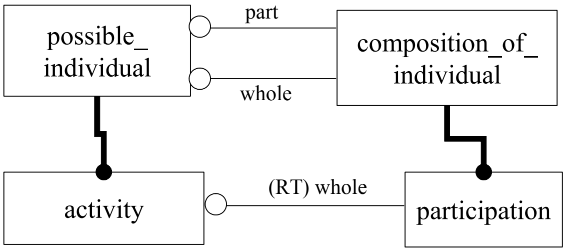


Figure 52 — Activity participation

EXAMPLE 1 Forming a plastic cup from a flat plastic blank using a hot moulding machine is an activity. The extent of the activity includes the plastic as its shape is changed and a temporal part of the machine being used. Figure 53 shows the spatio-temporal extent of the plastic involved in the "cup forming" activity. The forming activity causes the cup to come into existence and end the life of the plastic blank.

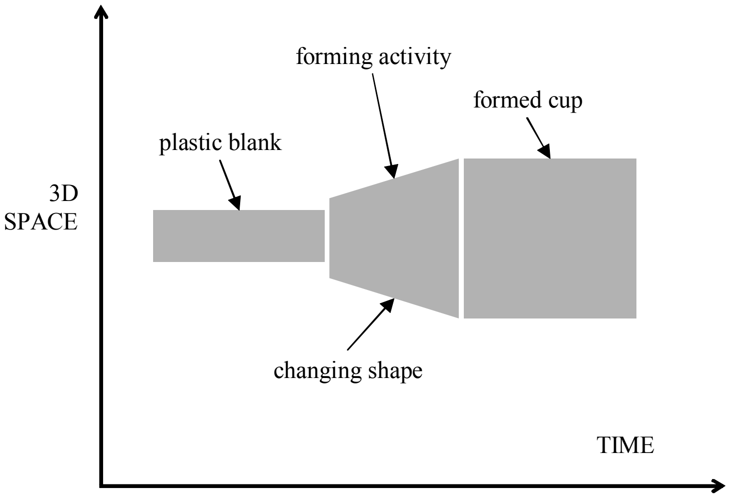


Figure 53 — Cup forming activity

The model instances necessary to show the participation of the molding machine, the plastic and a finished cup are shown in Figure 54.

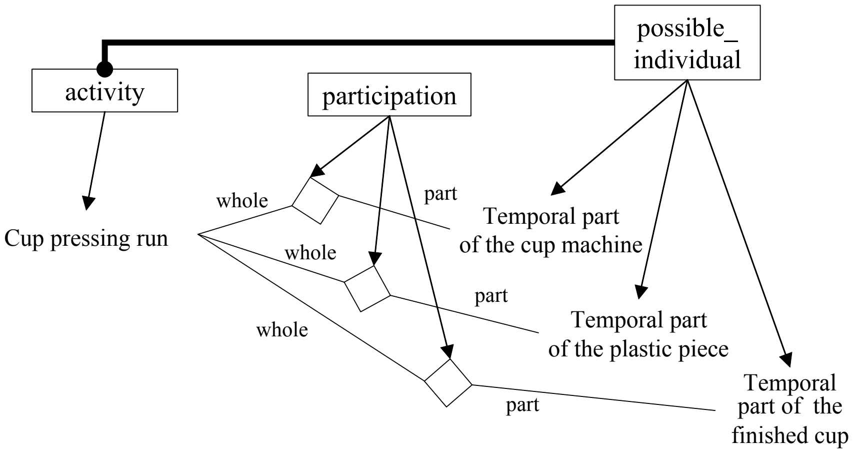


Figure 54 — Instance diagram of cup forming activity

Activities cause change that can be marked by **event**s (see 5.2.9.5). In Figure 53, two significant changes are shown: the start of the existence of the cup and the termination of the existence of the plastic blank. The model for cause is shown in Figure 55. **cause_of_event** is defined as a type of **relationship** (see 5.2.9.3).

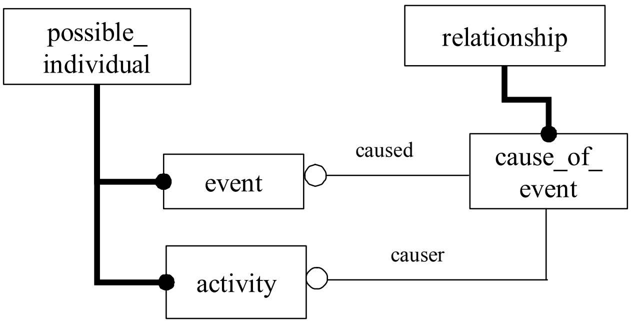


Figure 55 — Cause of event model

EXAMPLE 2 Figure 56 shows a particular cup pressing **activity** that causes the begin **event** of the cup and the ending of the plastic blank state.

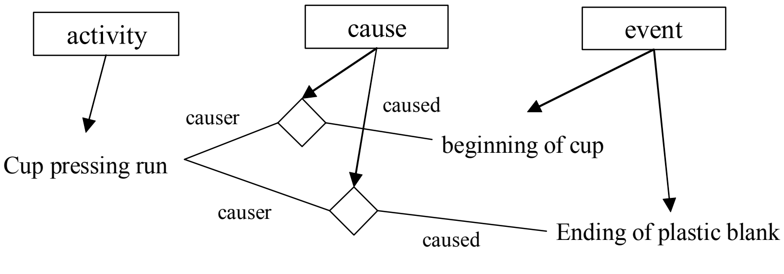


Figure 56 — Cup beginning caused by a cup pressing run

The model also allows **abstract_object**s and **possible_individual**s from the future or the past relative to the **activity** to be involved indirectly as shown in Figure 57 (see 5.2.9.6).

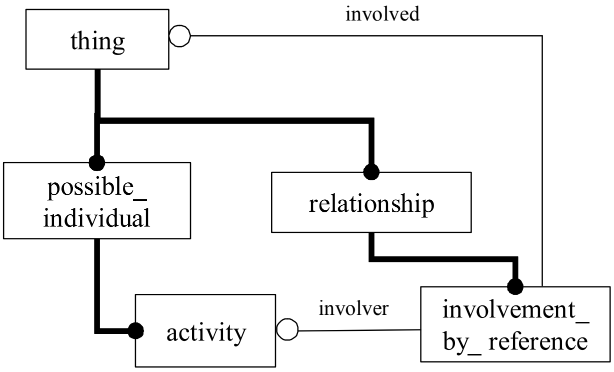


Figure 57 — Involvement by reference

EXAMPLE 3 A production **activity** may involve the specification for the individuals produced by the **activity**. The specification is a **class_of_individual** that has an **involvement_by_reference** relationship with the activity

Activities can also result in the recognition of abstract conditions (see 5.2.9.9). **recognition** is a relationship between an **activity** and a **thing** that indicates the **thing** has been recognised as a consequence of the **activity**. The elements of the model for **recognition** are shown in Figure 58 below.

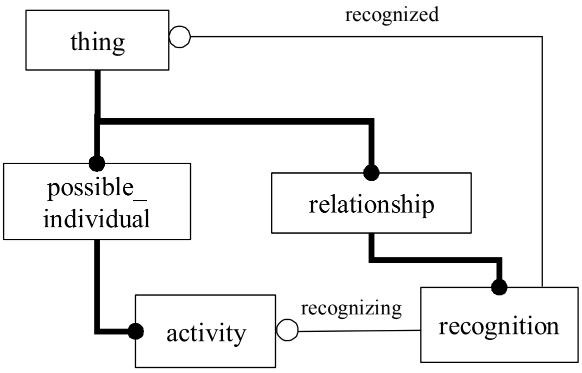


Figure 58 — Recognition by activity

EXAMPLE 4 Figure 59 shows a ship survey **activity** that recognizes that the ship has a 'Class A' classification.

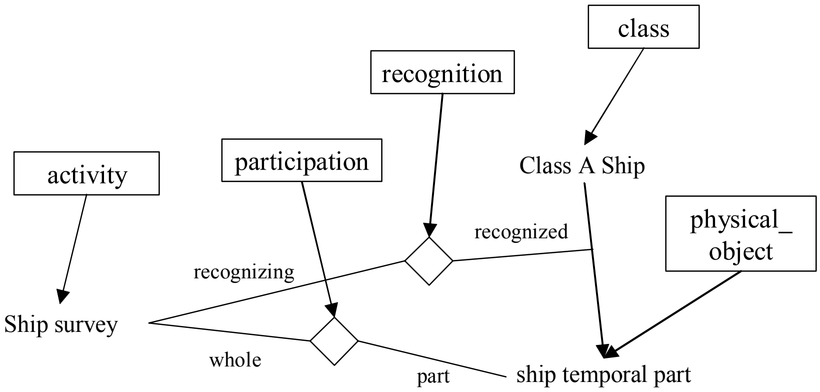


Figure 59 — Ship classification activity

4.7.18 Approval

approval is a **relationship** linking a **possible_individual** that could be a person, organisation or machine that grants the **approval** to the thing that is approved (see 5.2.23.1 and Figure 199). In this part of ISO 15926, only **relationships** can be approved as this gives a meaning to the **approval**. A **possible_individual** or a **class** is what it is, approving it is meaningless. It is approving its involvement with something else that has meaning.

The elements of the **approval** model are shown in Figure 60.

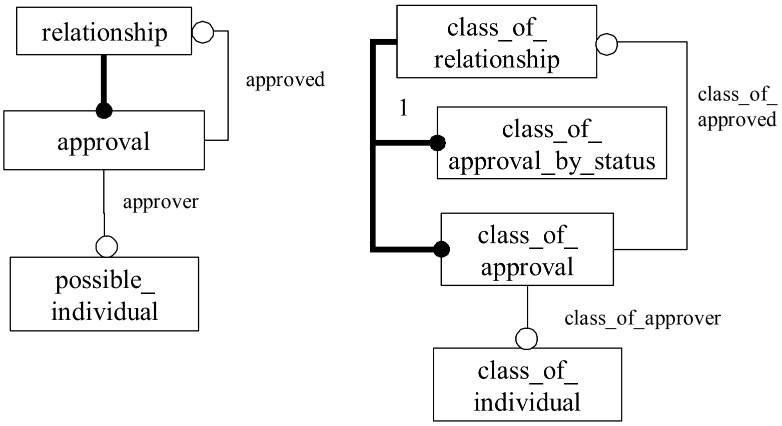


Figure 60 — Approval

The type of **approval** is given by classifying an **approval** relationship by a **class_of_approval_by_status** (see 5.2.23.3). **class_of_approval** enables rules on the classes of relationship that can be approved by types of individual (see 5.2.23.2).

EXAMPLE Figure 61 shows the **participation** of the raw plastic in the cup pressing **activity** being approved by the 'Production supervisor'. 'Approved' is a **class_of_approval_by_status**. 'Not approved' is also a **class_of_approval_by_status**.

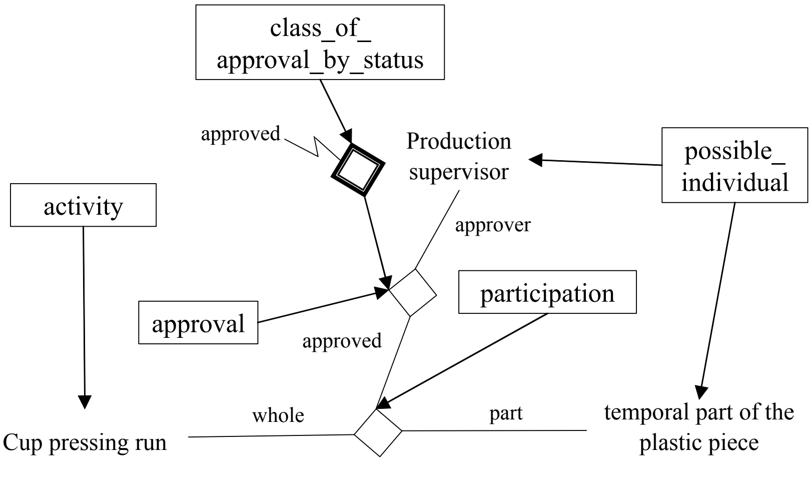


Figure 61 — Approval of cup raw material

4.8 Class

A **class** is a category, sort or a division of **thing**s with some common nature (see 5.2.2 and Figure 178).

classes have a basis for inclusion and exclusion.

Within this International Standard,

- inclusion within a **class** is referred to as class membership;
- **class**es correspond to non-well-founded sets. The characteristics of non-well-founded sets are further described in Annex D;
- the basis for inclusion or exclusion for a **class** can be given by a textual definition and or by its relationships.

4.8.1 Classification

classification is a **relationship** that indicates membership of a **class** (see 5.2.2.3 and Figure 178). Class membership implies that the member satisfies the conditions for inclusion and exclusion. **classification** is modelled as an explicit subtype of **relationship** as shown in Figure 62.

possible_individuals, **class**es, **relationship**s and **multidimensional_object**s may be classified. Classification is not transitive. Members of a **class** are not necessarily members of any **class** of the **class**.

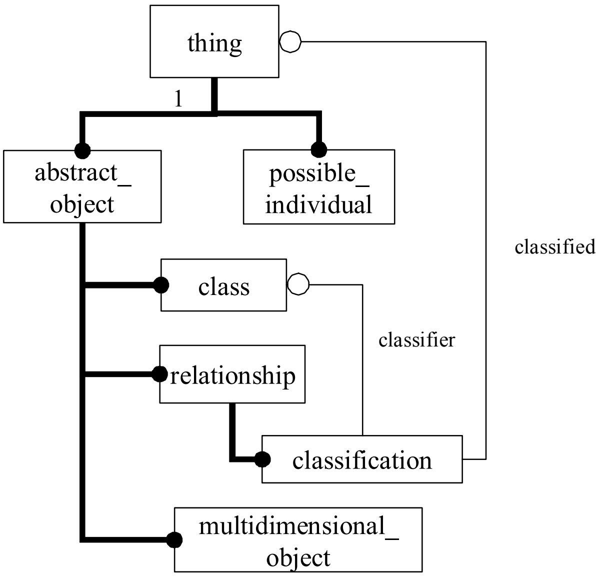


Figure 62 — Model of classification relationship

EXAMPLE 1 Figure 63 shows a category of **thing**s known as 'Pump' that is a **class**. Members of 'Pump' are **arranged_individual**s that enable pumping activity. The **arranged_individual** referred to as #1234 is a 'Pump'. The relationship that indicates #1234 is a member of 'Pump' **class** is a **classification**. In the case of the 'Pump' **class**, membership of it is not affected by whether or not the **arranged_individual** is a **whole_life_individual** or a temporal part of a **whole_life_indivdual**.

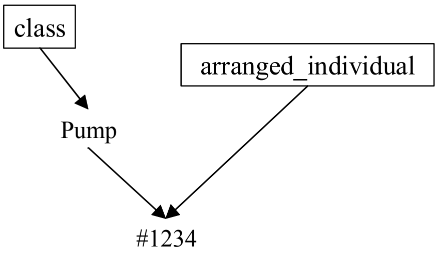


Figure 63 — Classification of pump

Classification allows something to be a member of many classes.

EXAMPLE 2 The **arranged_individual** referred to as #1234 is a ‘Pump’ and is also ‘operating’ as shown in Figure 64

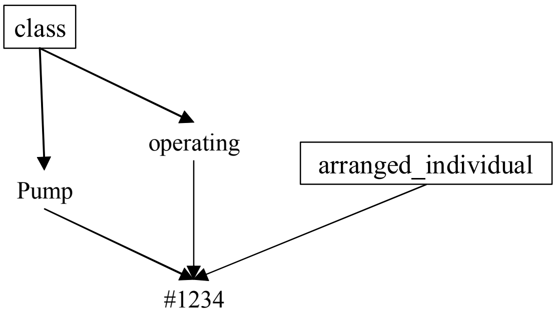


Figure 64 — Classification of operating pump

Many classifications that apply to a **whole_life_individual** also apply to all of the temporal parts of the **whole_life_individual** (a form of inheritance).

EXAMPLE 3 In Figure 65, #1234 is a temporal part of whole life ‘pump’ #PA01. The ‘pump’ classification rightly belongs to the **whole_life_individual** and is inherited by any and all of its temporal parts. However, if the **whole_life_individual** is not represented in the database, the classification can be recorded directly against the temporal part.

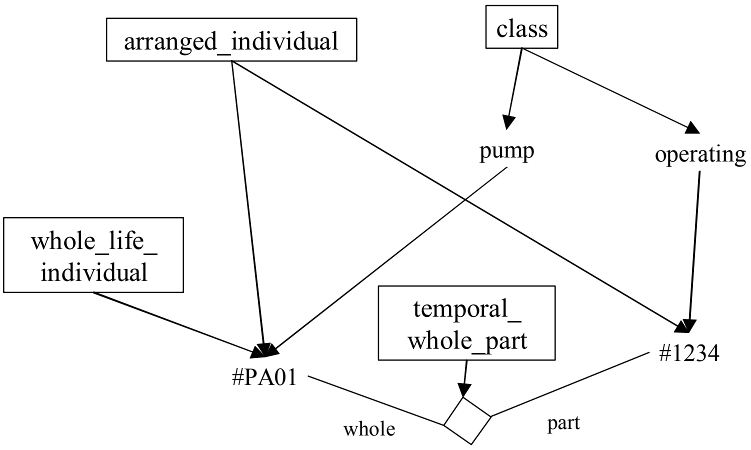


Figure 65 — Operating temporal part of a pump

4.8.2 Specialization

specialization is a type of **relationship** between two **class**es indicating the members of the subclass are members of the superclass as shown in Figure 66. **specialization** relationships are used to indicate a **class** is a subdivision of the membership of another **class** (see 5.2.2.4 and Figure 178).

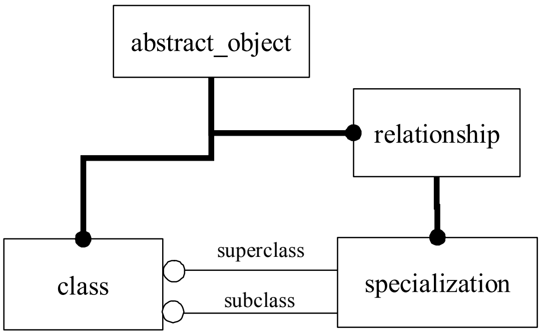


Figure 66 — Specialization relationship

Because the members of a subclass are members of the superclass, the subclass membership must conform to all the rules of membership of the superclass. The subclass is said to inherit the superclass rules.

EXAMPLE 1 Figure 67 showns a manufacturer's 'model 106' is a **class** that is a specialization of the "pump" **class** – all members of 'model 106' are also members of 'pump'.

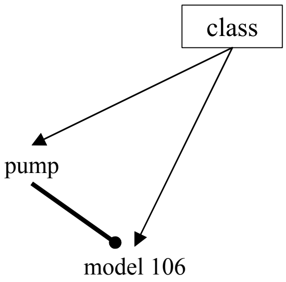


Figure 67 — Pump specialization

Specialization relationships are transitive. The members of a subclass of a subclass are members of the more general superclass.

EXAMPLE 2 In Figure 68 a manufacturer offers two options for their 'model 106' pumps, types 'A' and 'B'. Types 'A' and 'B' are both specializations of 'model 106'. A member of type 'A' is a member of 'model 106' and a member of 'pump'.

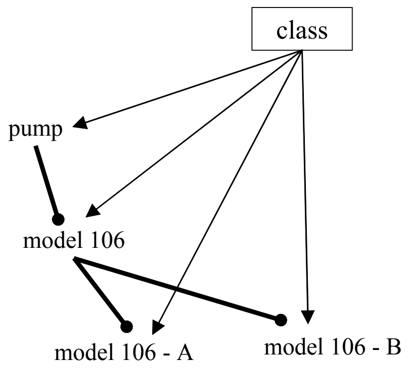


Figure 68 — Transitive specialization

Classification and specialization are fundamentally different. When **classification** refers to a **class** as the member, the members of the member **class** are not necessarily members of the classifying **class**.

EXAMPLE 3 In Figure 68, a member of 'model 106 – A' is not a **class**.

4.8.3 Types of class

The immediate subtypes of **class** recognized in this International Standard are shown in Figure 69.

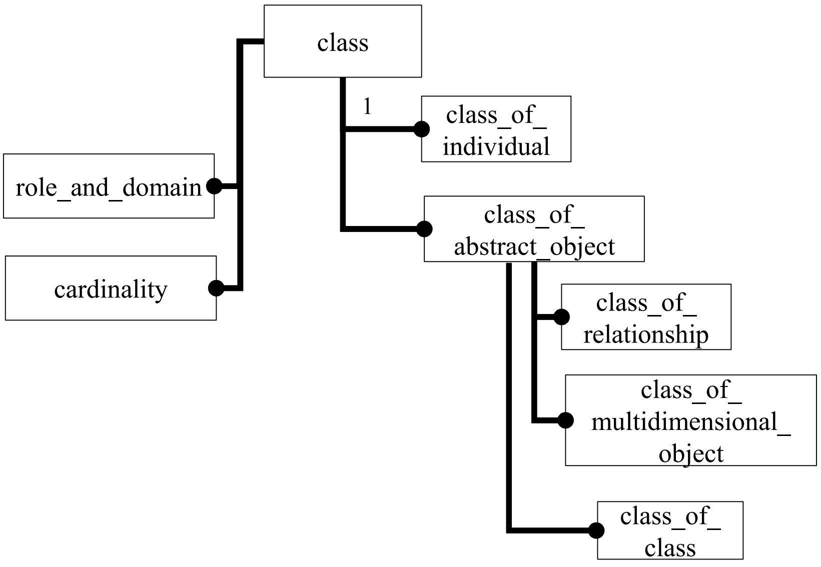


Figure 69 — Subtypes of class

4.8.3.1 Class of individual

A **class_of_individual** is a **class** whose members are space-time extensions i.e. **possible_individual**s (see 5.2.7 and Figure 183).

EXAMPLE 1 The class 'pump' is a **class_of_individual**.

EXAMPLE 2 The class 'red' is a **class_of_individual**; only space-time extensions can be 'red'.

4.8.3.2 Class of class

A **class_of_class** is a **class** whose members are **class**es (see 5.2.3 and Figure 179). **class_of_class** is a means of classifying the subdivisions of the membership of **class**. In other words, they can be used to identify types of subdivision.

EXAMPLE In Figure 70 the class 'colour' is a **class_of_class**. The **class**es 'red' and 'blue' are members of the **class_of_class** 'colour'.

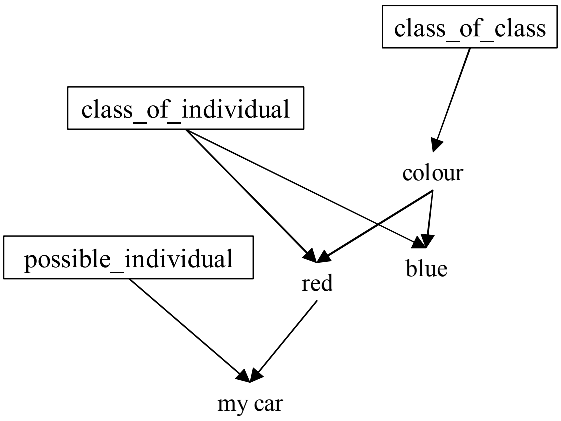


Figure 70 — Colour class of class

It should be noted that the EXPRESS subtypes of **class** defined in this part of ISO 15926 are instances of **class_of_class**, though the EXPRESS language specification does not allow this to be stated.

4.8.3.3 Class of relationship

class_of_relationship enables types of relationship to be recognised, often defining constraints in terms of the types of things that can participate in the member **relationship**s (see 5.2.12 and Figure 188).

This part of ISO 15926 defines some explicit subtypes of **class_of_relationship**, explicit in the sense that the roles of the relationships are explicitly defined as EXPRESS attributes. Other **class_of_relationship**, which are not explicitly defined, can be handled using **class_of_relationship_with_signature**. This is more fully described in 4.10.2.

The full list of explicit subtypes is specified in 5.2.12.2. The model for one explicit subtype **class_of_connection_of_individual** is shown in Figure 71.

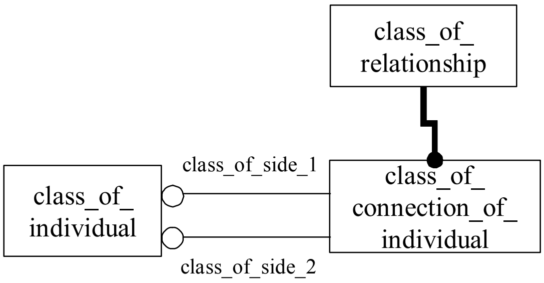


Figure 71 — Class of connection of individual

EXAMPLE In Figure 72, the **class_of_connection_of_individual** ‘connection of Type A shaft to seal’ indicates that members of the **class_of_individual** ‘Type A drive shaft’ are connected to members of ‘seal’. The **connection_of_individual** relationship, linking the shaft #5678 with the seal #1234, is a member of the **class_of_connection_of_individual** ‘Type A shaft seal’.

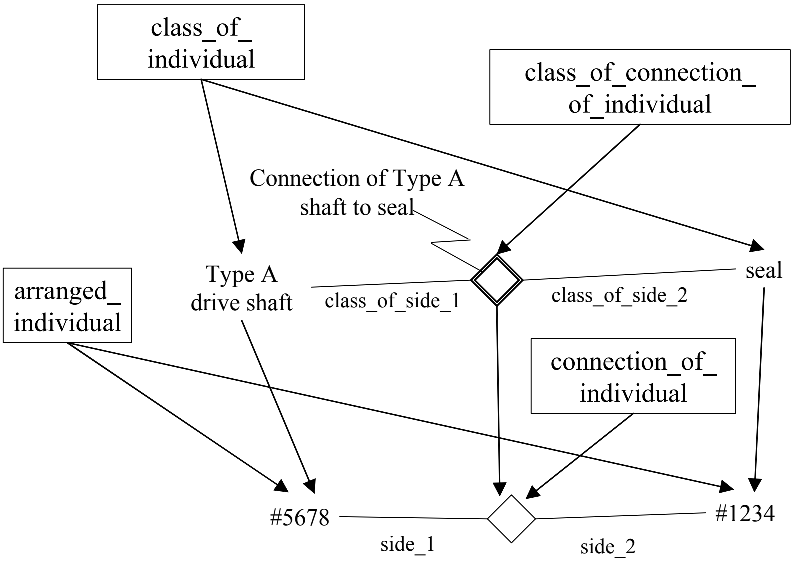


Figure 72 — Seal connected to Type A drive shaft

4.8.3.3.1 Cardinality constraints

Cardinality constraints of a **class_of_relationship** can be defined as attributes of the class **cardinality** (see 5.2.13.5, 5.2.13.1 and Figure 189). Figure 73 shows the model diagram for this. The cardinality attributes work in the same way as the cardinality constraints often shown on Entity-Relationship diagrams. The number of member relations in the **class_of_relationship** that link from one member at the first end to members at the other end is constrained by the minimum and maximum values. If no maximum cardinality is set, no limit shall be assumed and, if no minimum cardinality is set a limit of zero shall be assumed.

The **end_1_cardinality** and **end_2_cardinality** attributes shall be applied to the role attributes in the order of their declaration in the EXPRESS definition of the **class_of_relationship**.

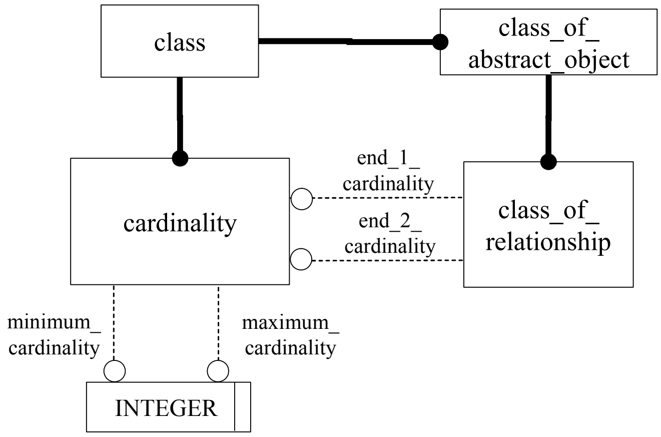


Figure 73 — Cardinality constraints for classes of relationship

Cardinality constraints may be applied to the members of the explicit subtypes of **class_of_relationship**.

EXAMPLE A Type A drive shaft requires to be located by two seals. Figure 74 shows the cardinalities for this case. A shaft may be connected to zero, one or two seals, and a seal may be connected to zero or one shaft. The zero and one cardinality for seals of a shaft recognises that at any time, one or more of the seals may be removed for maintenance purposes.

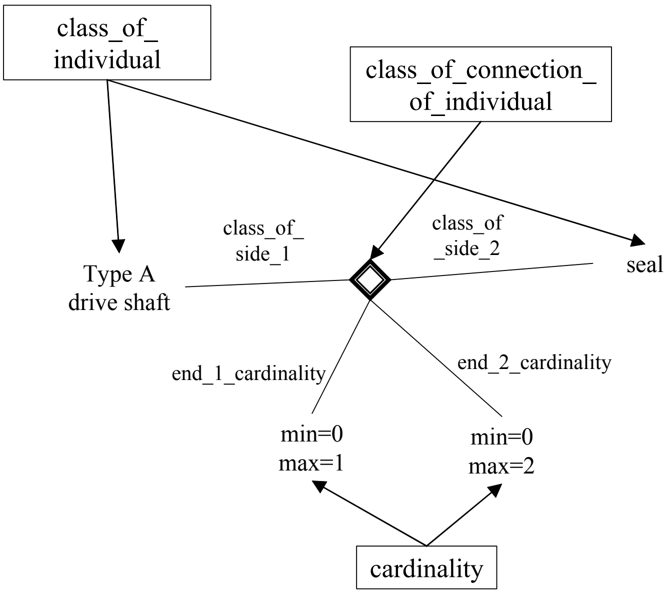


Figure 74 — Type A drive shafts may connect to up to two seals

The use cardinality with **class_of relationship_with_signature** is described in 4.10.3.

4.8.3.3.2 Class of relationship symmetry

Most of the explicit subtypes of **class_of_relationship** define constraints on the member **relationship**s by referring to **class**es. In the case of **class_of_connection_of_individual**, the roles **class_of_side_1** and **class_of_side_2** both indicate that the member **relationship**s refer to participation of members of the indicated classes. These cases are described as symmetric **class_of_relationship**.

Not all **class_of_relationship** are symmetric. **class_of representation_of_thing**, shown in Figure 75, is such that the member relationships always refer to the same **thing** in the represented role and where this is a **class**, it means the **class** and not its members. Cases such as these are described as asymmetric **class_of_relationship**s.

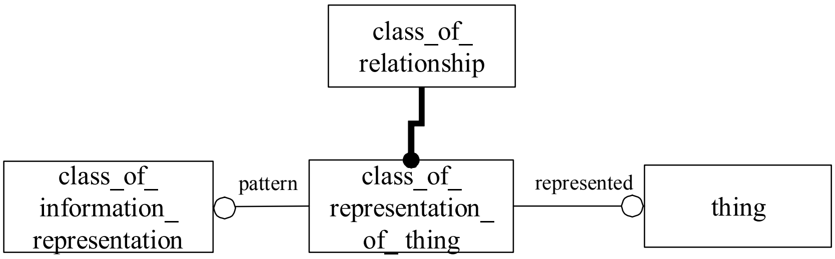


Figure 75 — An asymmetric class of relationship

A symmetric **class_of_relationship** can be further constrained to make an asymmetric **class_of_relationship**. The elements of the model for this are shown in Figure 76. A **class_of_relationship** can be both a **class_of_connection_of_individual** and, for example, a **class_of_relationship_with_related_end_2**. The related attribute further constrains the domain of **class_of_side_2** to be one **thing** only. The entity types **class_of_relationship_with_related_end_1** and **class_of_relationship_with_related_end_2** in that order, shall constrain the **class_of_relationship** roles in the order given in the EXPRESS definition of the model.

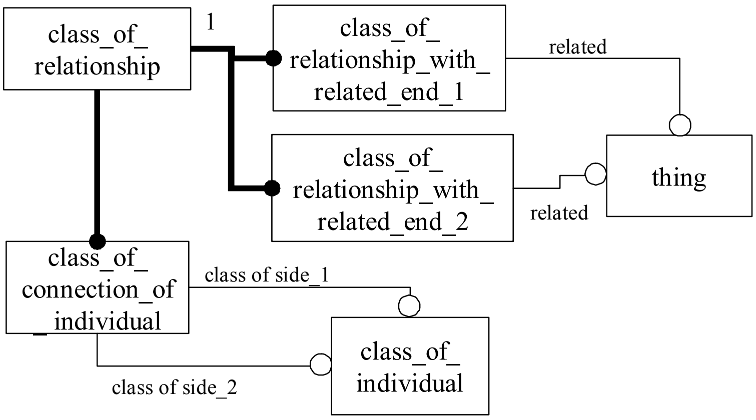


Figure 76 — Constraining a symmetric class of relationship

EXAMPLE Figure 77 shows a whose members are all the **connection_of_individual** relationships that link a seal to the particular shaft #5678. The known members of this class would give the history of seals connected to the particular shaft. The **class_of_connection_of_individual** shown in Figure 77 is a specialization of the **class_of_connection_of_individual** 'Type A shaft seal' shown in Figure 72.

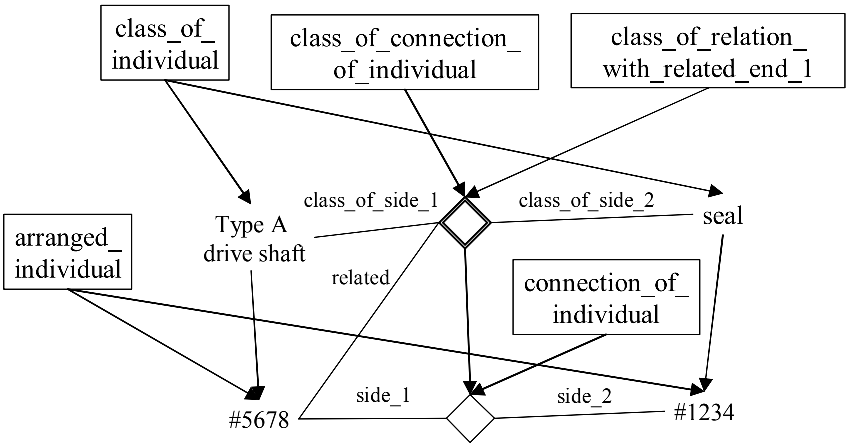


Figure 77 — Seals connected to a particular shaft

In the example the attribute related is a specialization of the attribute **class_of_side_1**, but the rules of the EXPRESS language prevent this being shown.

4.8.4 Class of individual

A **class_of_individual** is a class whose members are space-time extensions i.e. **possible_individuals** (see 5.2.7 and Figure 183).

The explicitly defined subtypes of **class_of_individual** are shown in Figure 78.

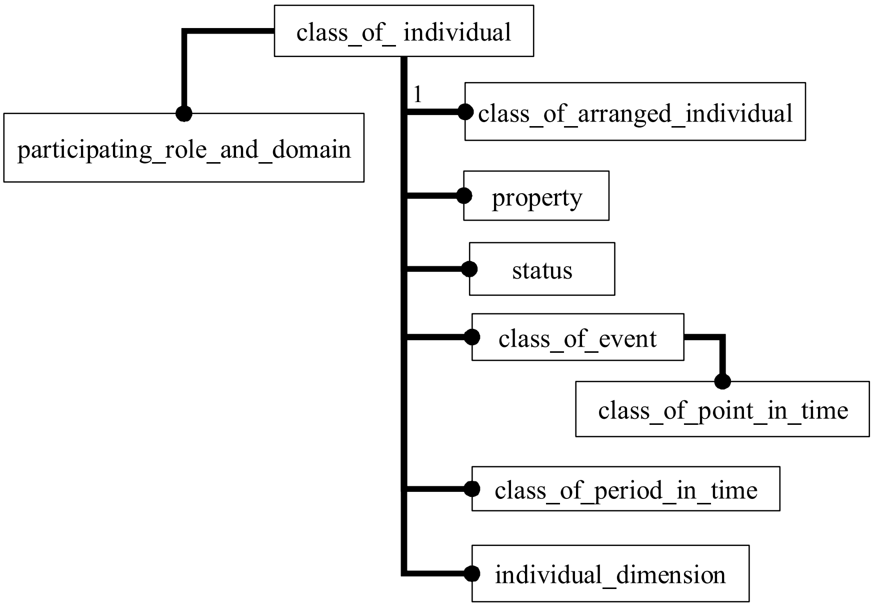


Figure 78 — Subtypes of class_of_individual

Rules regarding the composition of members of a **class_of_individual** may be specified using **class_of_composition_of_individual** (see 5.2.7.5). The model of **class_of_composition_of_individual** is shown in Figure 79. Explicit subtypes are defined for **class_of_temporal_whole_part**, **class_of_participation**, **class_of_arrangement_of_individual** and **class_of_assembly_of_individual** (see 5.2.7.12, 5.2.10.5, 5.2.7.1 and 5.2.7.2).

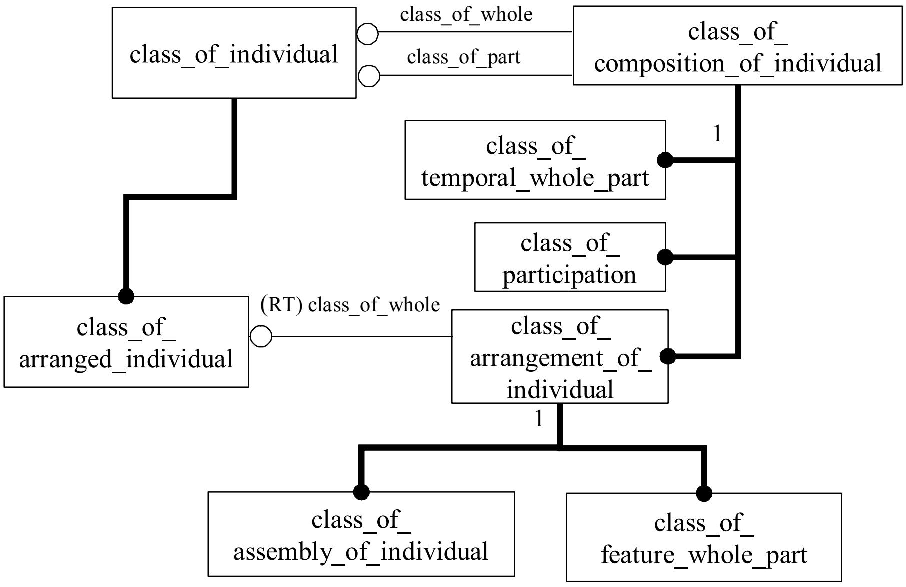


Figure 79 — Class of composition of individual and subtypes

A **class_of_composition_of_individual** can be used to express a condition on the members of the class of whole, constraining them to have parts that are members of the class of part.

EXAMPLE In Figure 80 centrifugal pump is a **class_of_arranged_individual**. The **class_of_assembly_of_individual** 'Pump impeller' records that a 'centrifugal pump' has an 'impeller' as a part.

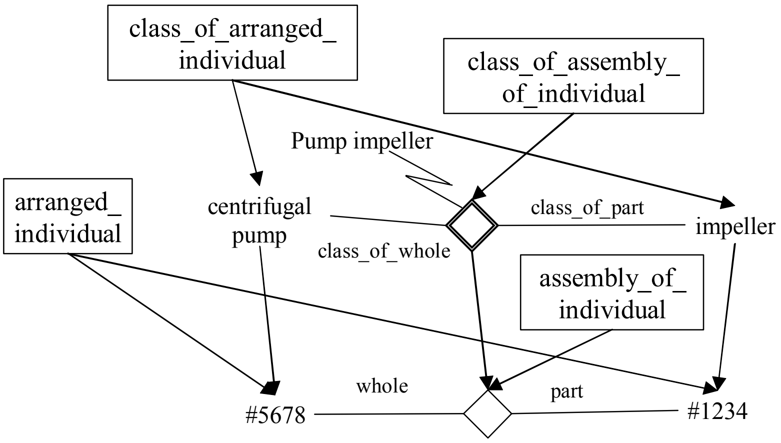


Figure 80 — Composition of centrifugal pumps

4.8.4.1 Class of arranged individual

arranged_individuals are space-time extensions whose parts have a particular arrangement or role in the whole (see 5.2.6.2). A **class_of_arranged_individual** is a subdivision of **arranged_individual**s. Explicit subtypes are defined covering:

- arrangements based on organization of material;
- arrangements based on information presentation and representation;
- complex arrangements that align with useful descriptions of every day objects.

4.8.4.1.1 Organization of material classes

Explicit subtypes of **class_of_arranged_individual** recognizing different types of material orgnization are defined as follows (see 5.2.8 and Figure 184):

- **class_of_sub_atomic_particle**;
- **class_of_atom**;
- **class_of_molecule**;
- **class_of_compound**;
- **crystalline_structure**
- **phase**
- **class_of_particulate_material**;
- **class_of_composite_material**;
- **class_of_functional_object**;
- **class_of_biological_matter**.

The model for the material organization classes is shown in Figure 81.

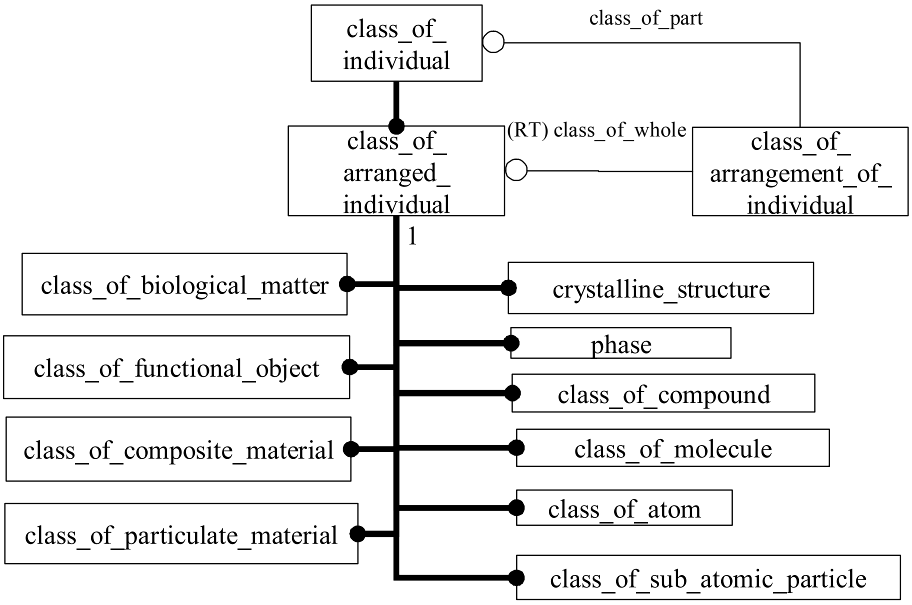

class_of_individual
class_of_part
class_of_arranged_individual
(RT) class_of_whole
class_of_arrangement_of_individual
1
class_of_biological_matter
crystalline_structure
phase
class_of_functional_object
class_of_compound
class_of_composite_material
class_of_molecule
class_of_atom
class_of_particulate_material
class_of_sub_atomic_particle

Figure 81 — Classes of arranged individual for material structure

The material organization types reflect increasing levels of compositional arrangement, starting with the subatomic particles culminating in functional objects and biological matter.

EXAMPLE 1 'Hydrogen atom' is a **class_of_atom**. Members of 'Hydrogen atom' are individual atoms each of which are an arrangement of a 'neutron', 'proton', and an 'electron' where 'neutron', 'proton' and 'electron' are members of **class_of_sub_atomic_particle**.

In some cases, members of a **class** at each level are aggregates of members of one or more **class**es of the next level down. In this part of ISO 15926, large-scale aggregates of molecules or atoms are defined as compounds.

EXAMPLE 2 An aggregate of hydrogen molecules, perhaps forming a gas, is member of the 'Hydrogen' **class_of_compound**.

EXAMPLE 3 'Water' is a **class_of_compound** that is an arrangement of H_2O molecules. The levels of arrangement are illustrated in Figure 82. Figure 83 shows an instance diagram of water as an arrangement of H_2O molecules.

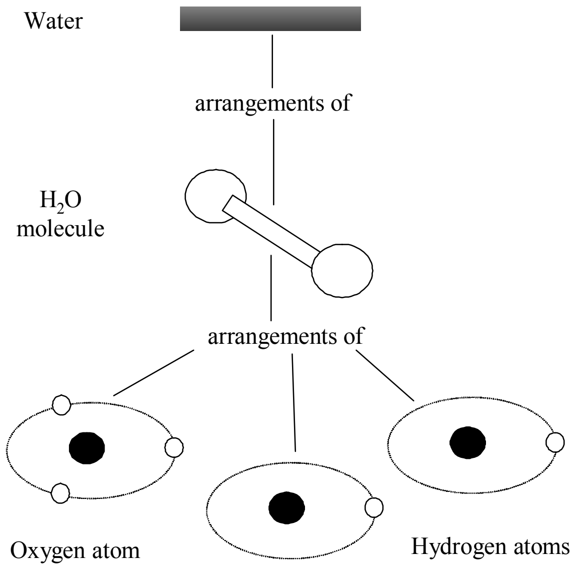


Figure 82 — Levels of arrangement of for water

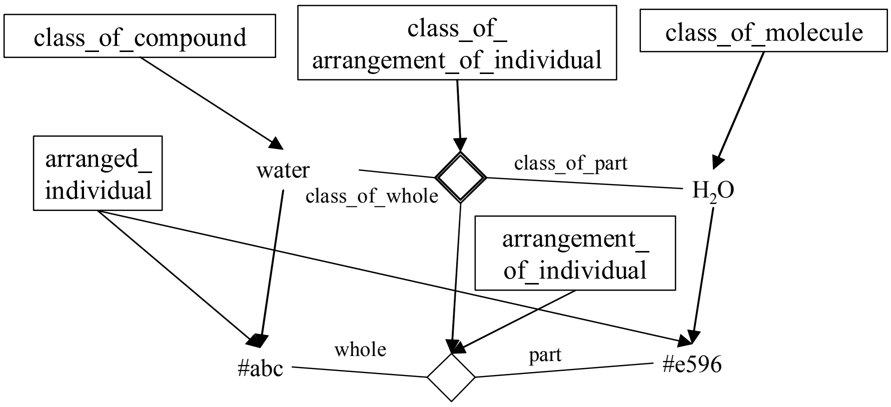


Figure 83 — Arrangement of H_2O molecules

Two other types of **class_of_arranged_individual** are defined for **crystalline_structure** and **phase**. These qualify material aggregations at the level of compound.

EXAMPLE 4 A cloud of hydrogen gas is a member of the 'hydrogen' **class_of_compound**, and a member of the **phase** class 'gas'.

EXAMPLE 5 A diamond is an aggregate of carbon atoms that is a solid with a particular **crystalline_structure**.

Particulate and composite material classes are further levels of arrangement, where the parts are compounds.

EXAMPLE 6 'Sand' is a **class_of_particulate_material** whose parts are constrained to be members of the 'silica compound' class.

EXAMPLE 7 'Fibre glass' is a **class_of_composite_material** whose parts are constrained to be members of the 'glass fibre mat composite' and 'resin compound' classes.

Functional objects are considered to be the highest level of organisation. Here, the member **arranged_individual**s are constrained by being suitable for some function or purpose. Other considerations that may make the function realisable such as weight, size, shape or material are excluded. The combination of purpose and form are recognised by the complex arrangement classes described in 4.8.4.1.2.

EXAMPLE 8 'Cup' and 'pump' are examples of **class_of_functional_object**.

EXAMPLE 9 'Centrifugal pump' is not a **class_of_functional_object**, centrifugal being a form or design of a pump function.

4.8.4.1.2 Complex arrangements

Complex arrangement classes enable more detailed types of individual to be recognised (see Figure 184). They are often intersections of many other classes combining aspects of material, shape, and property. The explicit types for these complex arrangements are shown in Figure 84.

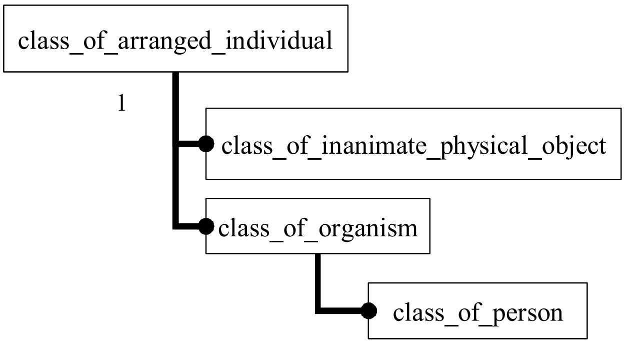

class_of_arranged_individual
1
class_of_inanimate_physical_object
class_of_organism
class_of_person

Figure 84 — Complex classes of arranged individual

EXAMPLE 'Plastic cup' is a **class_of_inanimate_physical_object** that is the intersection of the **class_of_functional_object** 'cup' and **class_of_compound** 'plastic' as shown in Figure 85.

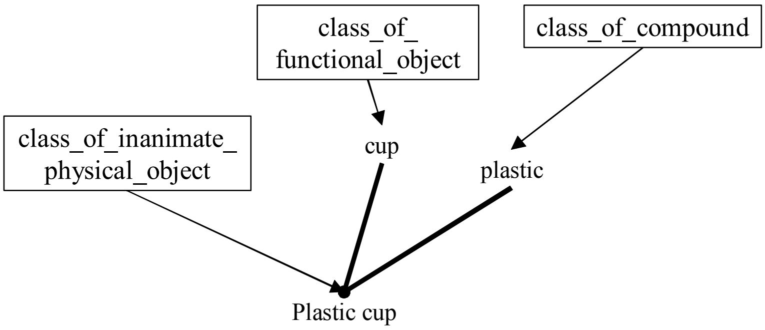


Figure 85 — Class of inanimate physical object

4.8.4.1.3 Information classes

Representation of meaning using symbols depends on using consistent, recognisable patterns. Patterns are classes. A particular writing or rendering, say on a piece of paper or on a video screen that can be observed with our senses, is a possible individual that is a member a pattern class.

In this part of ISO 15926, a **class_of_information_representation** identifies a pattern used to represent information (see 5.2.17.4 and Figure 193).

The rendered patterns often have many presentational variations such as colour, font, size, and weight. **class_of_information_presentation** describes these variations (see 5.2.8.10 and Figure 184).

Members of **class_of_information_object** are the combinations of the recognizable patterns and their presentation styles (see 5.2.8.9). The model elements for this are shown in Figure 86.

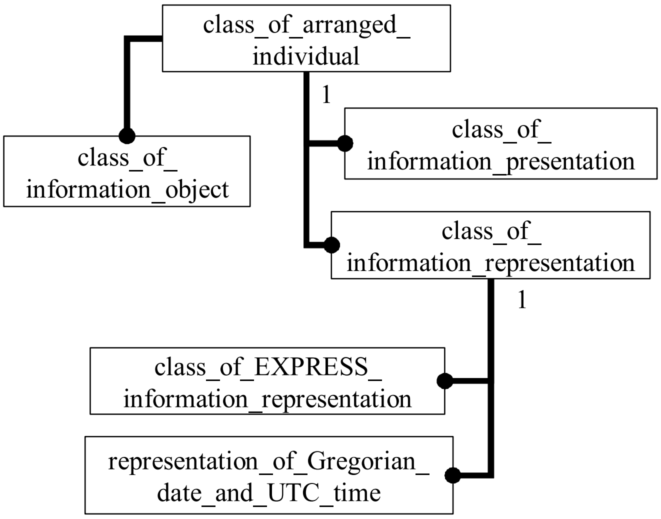

class_of_arranged_ individual
1
class_of_ information_presentation
class_of_ information_object
class_of_ information_representation
1
class_of_EXPRESS_ information_representation
representation_of_Gregorian_ date_and_UTC_time

Figure 86 — Information classes of arranged individual

EXAMPLE Figure 87 shows an **arranged_individual** #smith that is a member of the **class_of_inanimate_physical_object** 'label' and a member of the **class_of_information_object** '`smith` in 24 pt Times New Roman bold' . The class '`Smith` in 24 pt Times New Roman bold' is an intersection of the **class_of_information_representation** 'smith' and the presentation classes 'Times new roman', 'bold' and '24pt'. The physical aspects of the label substrate are not shown.

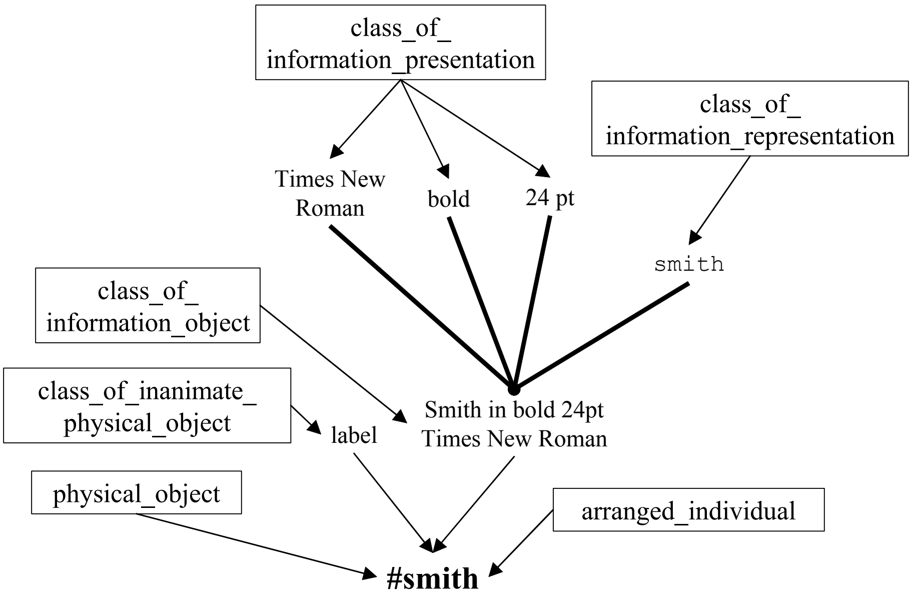


Figure 87 — Class of information object

The literal character patterns defined by ISO 10303 EXPRESS for text strings, reals, integers, binary, logical and Boolean and the ISO 8061 time representation are defined as explicit subtypes of **class_of_information_representation**.

4.8.4.2 Representation

4.8.4.2.1 Signs and patterns

Representation is the use of signs and patterns as information. A sign is a role of a **possible_individual** – i.e. a space-time extension. Signs can be any individual and can represent any **thing**. The representation model for signs is shown in Figure 88.

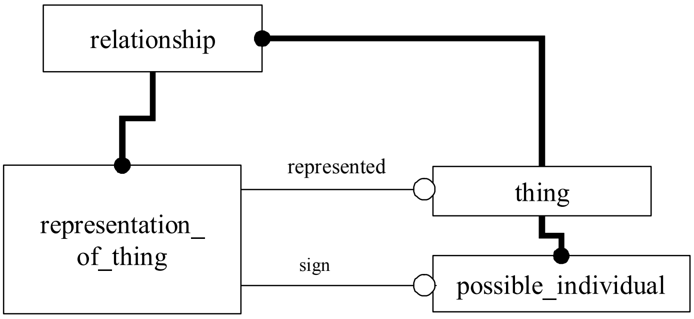


Figure 88 — Representation of thing

representation_of_thing is a **relationship** that indicates that a **possible_individual** is a sign for something else (see 5.2.16.4 and Figure 192).

Patterns are types or classes of signs, the pattern being the repeatable nature of the member signs. Signs that are members of the same pattern are often used to represent the same thing. So the pattern "Joe Smith" wherever, whenever and how rendered usually refers to the person. The model for pattern representation is shown in Figure 89 (see also 5.2.17.5 and Figure 193).

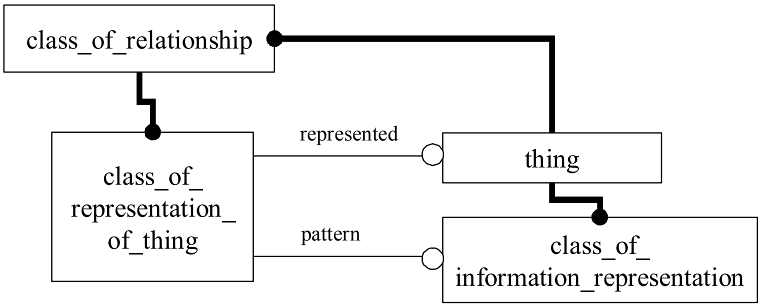


Figure 89 — Class of representation of thing

EXAMPLE Figure 90 shows Smith the person as the **actual_individual** #3578 linked to the sign #smith by a **representation_of_thing** relationship. The sign #smith is an inanimate **physical_object** that is a member of the 'Smith' pattern. The **representation_of_thing** relationship is a member of the **class_of_representation_of_thing** linking the pattern 'Smith' to #3578.

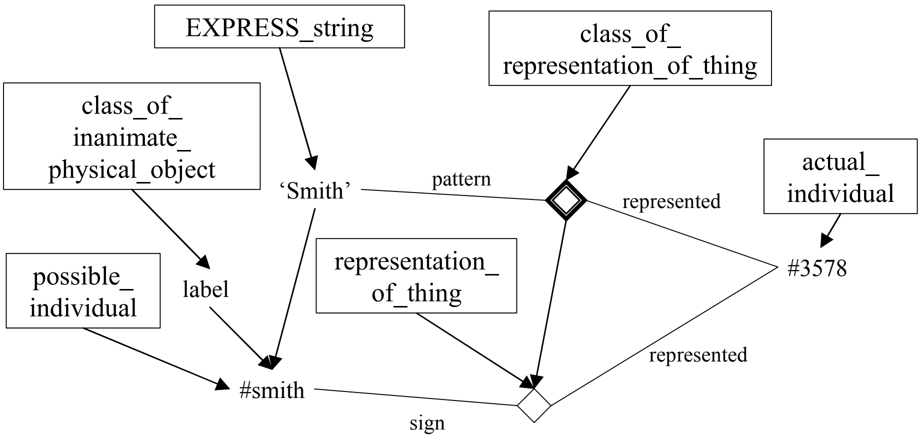


Figure 90 — Representation of #3578

4.8.4.2.2 Identification, description and definition

identification, **description** and **definition** are all types of representation that apply to alphanumeric, picture and sound signs (see 5.2.16.1, 5.2.16.2 and 5.2.16.3). Because individuals cannot be defined, they are what they are, **definition**s are restricted to **class**es as shown in Figure 91.

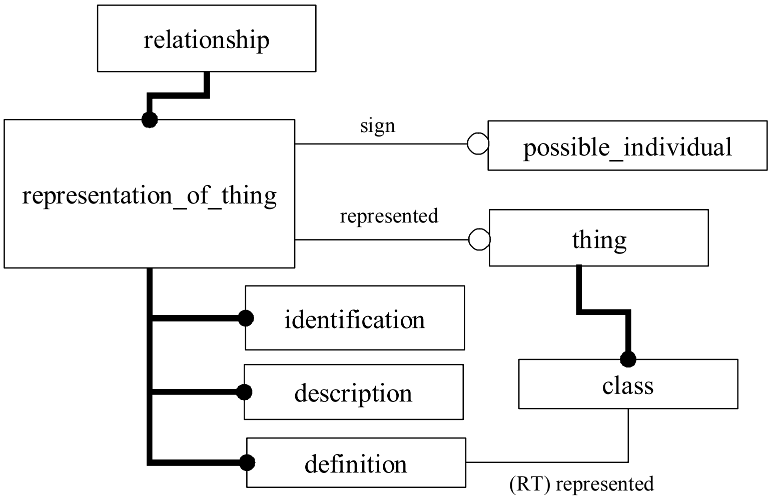


Figure 91 — Identification, description and definition

In process plant related activities **identification**, **description** and **definition**s are more often declared at the pattern level, applying to all signs of the pattern as shown in Figure 92 (see 5.2.17.3, 5.2.17.2, and 5.2.17.1).

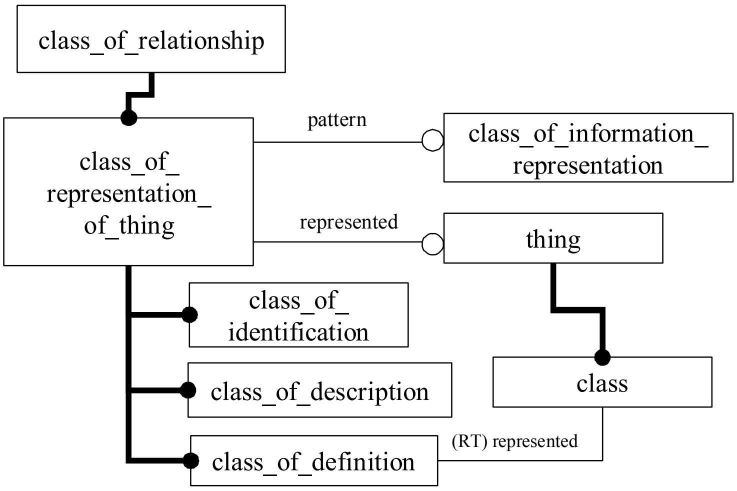


Figure 92 — Class of Identification, description and definition

EXAMPLE The **arranged_individual** known as 'My pump' has been given an **identification** pattern of 'AC-1234'. The **identification** sign 'AC-1234' appears on the 'name plate' of the pump, as shown in Figure 93. The composition relationship that would state that the 'name plate' is part of the pump is not shown.

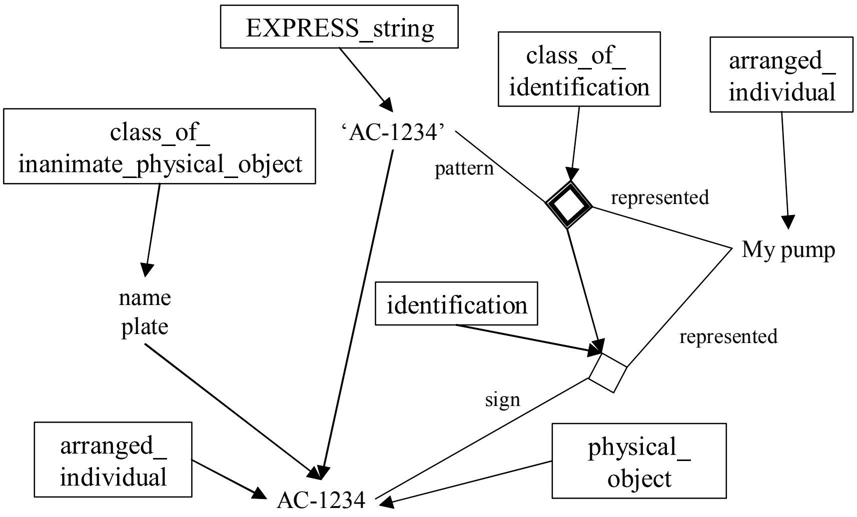


Figure 93 — Pump identification

4.8.4.2.3 Use and responsibility of representation

The use of certain signs and patterns as representations for particular things is discretionary, and may be restricted to certain people and organisations. The same pattern may be used to represent different things by different people or organisations and, a particular thing may have several representation patterns assigned and used by different organisations.

This part of ISO 15926 distinguishes responsibility from general use. Responsibility is used to indicate the person or organisation that took the decision to assign the sign or pattern to the thing that it represents (see 5.2.16.5 and Figure 192). Use indicates that an organisation or person uses the representation in their activities (see 5.2.16.6 and Figure 192).

The model for use and responsibility of representation is shown in Figure 94.

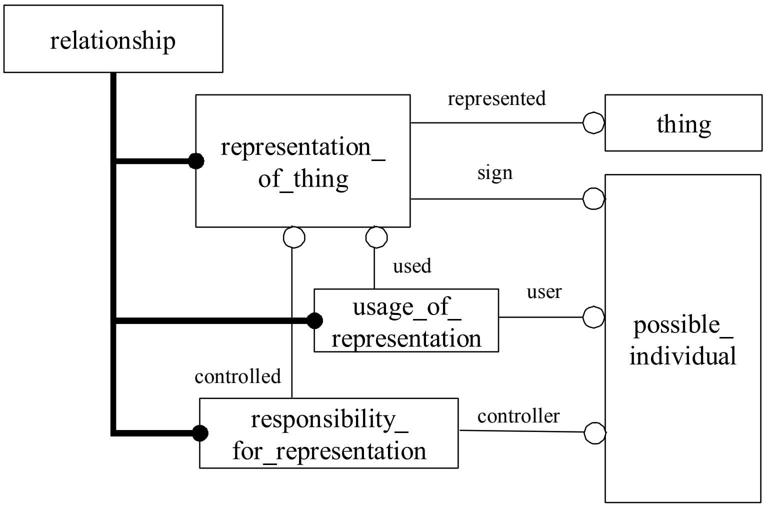


Figure 94 — Use and control of representation

In process plant related activities use and responsibility of representation are more often declared at the pattern level, applying to all signs of the pattern as shown in Figure 95 (see 5.2.17.7, 5.2.17.8 and Figure 193).

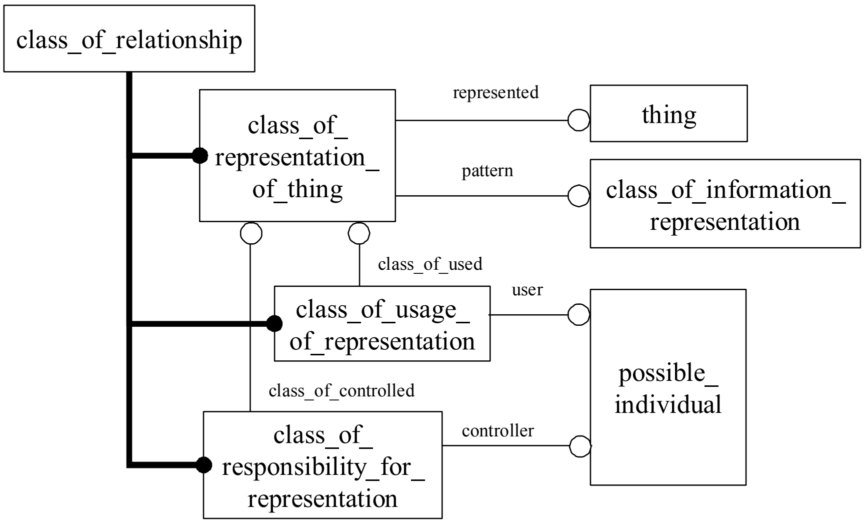


Figure 95 — Use and control of class of representation

EXAMPLE Figure 96 showns the **identification** pattern ‘AC-1234’ for ‘My pump’ issued by the ‘XYZ Co’ and used by the ‘ABC Oil Co’.

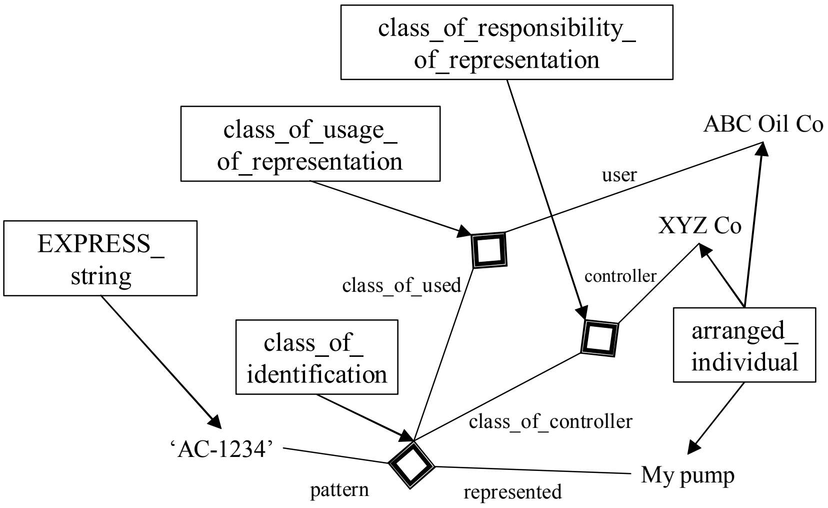


Figure 96 — XYZ Co product identifiers

4.8.4.2.4 Classes of pattern

Patterns may be further abstracted to define rules for the types of thing the member patterns may represent and to define rules constraining the composition of member patterns (see 5.2.19 and Figure 195). The classes of pattern are defined as **class_of_class_of_information_representation**, shown in Figure 97. Explicit subtypes are given for **language**, **representation_form**, and **document_definition** (see 5.2.19.9, 5.2.19.10 and 5.2.19.11).

EXAMPLE 1 The template used for an engineering data sheet is a **document_definition**.

EXAMPLE 2 ‘Hexadecimal’ is a **representation_form** that is a specialization of ‘text’ that is also a **representation_form**.

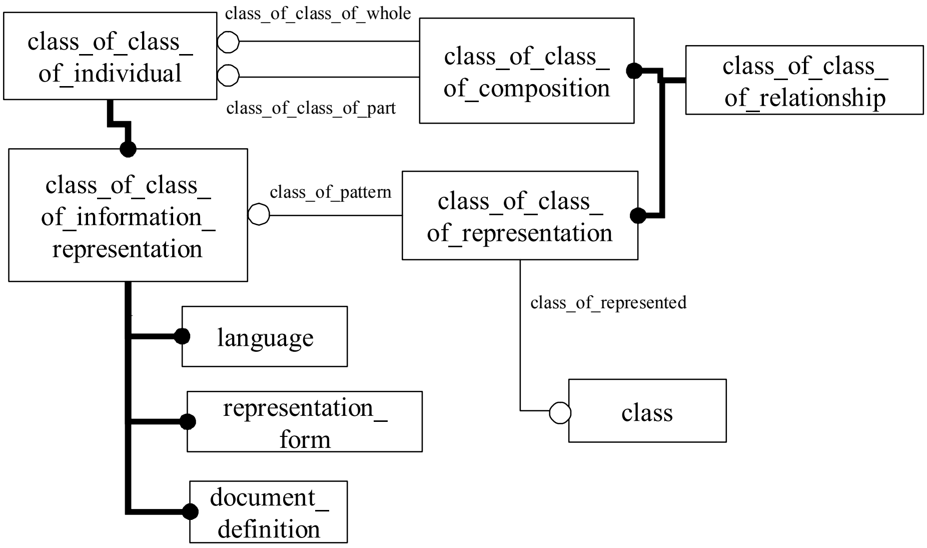

class_of_class_of_individual
class_of_class_of_whole
class_of_class_of_part
class_of_class_of_composition
class_of_class_of_relationship
class_of_class_of_information_representation
class_of_pattern
class_of_class_of_representation
class_of_represented
language
representation_form
document_definition
class

Figure 97 — Class of class of information representation

EXAMPLE 3 Figure 98 represents that an ISO 10303-21 (Part 21) [1] file contains a record that represents the **functional_physical_object** P101. ‘P21 file’ is a **representation_form** that has ‘P21 record’s as parts.

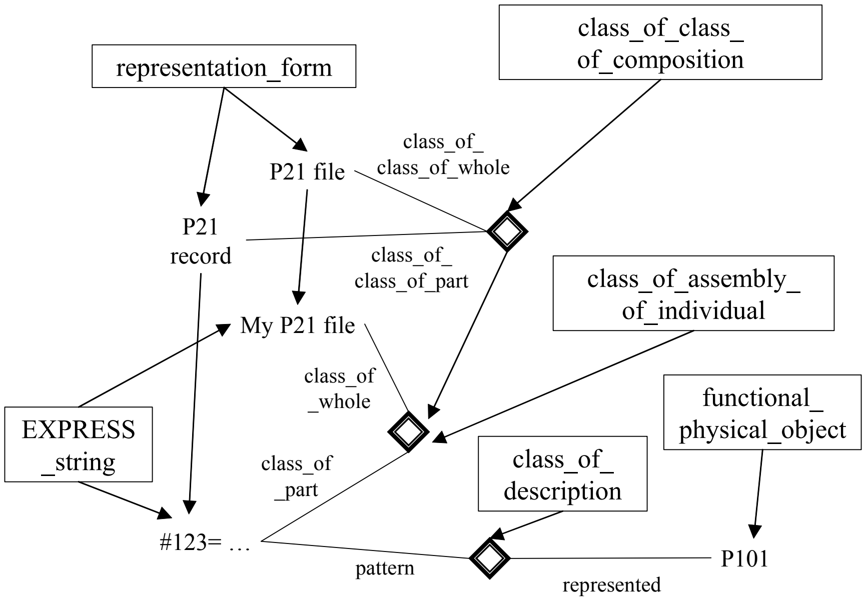


Figure 98 — Part 21 representation

People and organisations may use and or control certain classes of pattern to represent certain types of thing. The model for this is shown in Figure 99. **class_of_class_of_usage_of_representation** and **class_of_class_of_responsibility_for_representation**, both subtypes of **class_of_class_of_relationship**, are defined for these purposes (see 5.2.19.7 and 5.2.19.8).

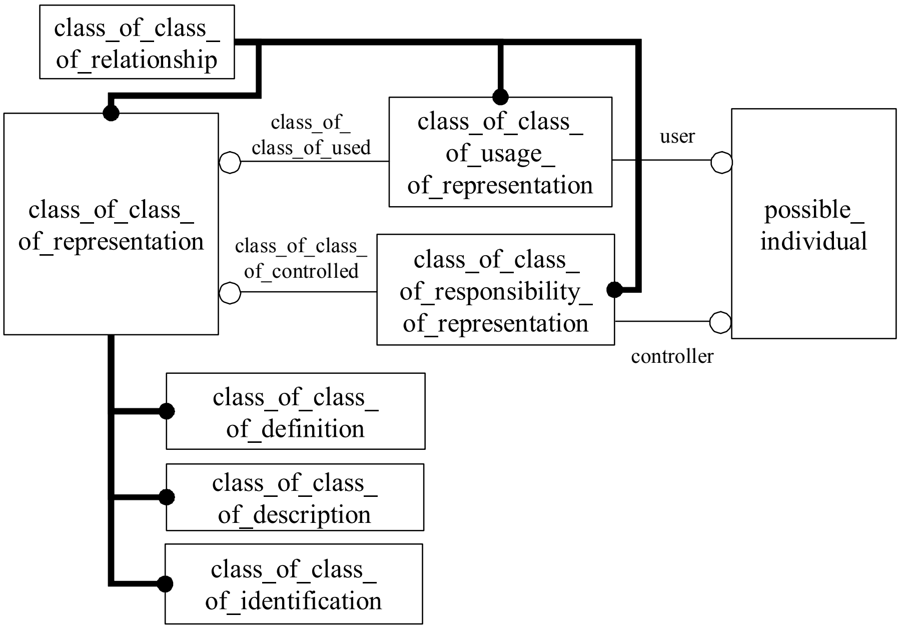


Figure 99 — Class of class of usage and responsibility of representation

EXAMPLE 4 The pump data sheet format designed and maintained by the 'XYZ Co' is a **document_definition** that describes all types of 'XYZ Co' designed pumps. The 'type 24 pump' is described by the 'Type 24 pump data sheet', which is a **class_of_information_representation**. Note the 'Type 24 pump data sheet' described the class of pumps, not any particular pump member. The content of the data sheet is not prescribed in the example shown in Figure 100.

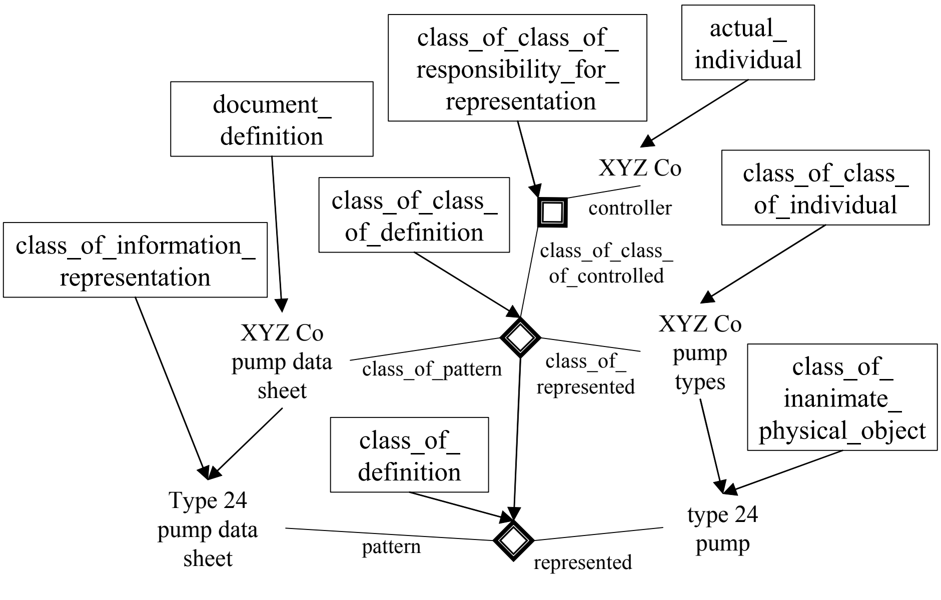


Figure 100 — XYZ Co pump data sheets

4.8.4.2.5 Namespace patterns

Some naming conventions prescribe the use of patterns that have constant suffixes or prefixes as identifiers (see 5.2.20 and Figure 196). Such classes of patterns can be defined using **class_of_namespace** (see 5.2.20.2). **class_of_namespace** constrains a class of pattern to have a constant part, either the left-most part or the right-most part (see 5.2.20.1 and 5.2.20.3). The model for this is shown in Figure 101.

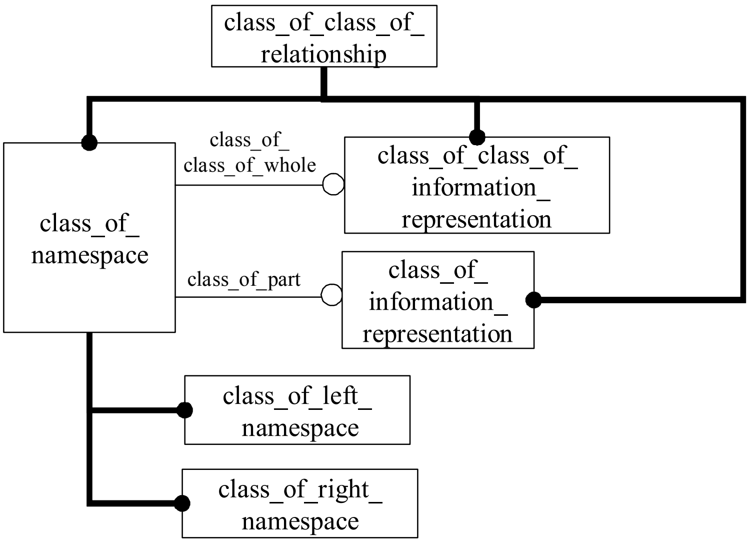

class_of_class_of_relationship
class_of_namespace
class_of_class_of_whole
class_of_class_of_information_representation
class_of_part
class_of_information_representation
class_of_left_namespace
class_of_right_namespace

Figure 101 — Class of namespace

EXAMPLE 1 As shown in Figure 102, the set of nozzle identifiers for vessel ‘V1’ are ‘V1:’ followed by a nozzle identifier. The ‘V1’ identifiers are members of the **class_of_class_of_information_representation** shown as ‘V1:*’. A **class_of_left_namespace** indicates that the members of ‘V1:*’ all start with ‘V1:’ followed by a nozzle identifier from the enumerated set “‘N1’ , ‘N2’, ‘N3’, ‘N4’ …”

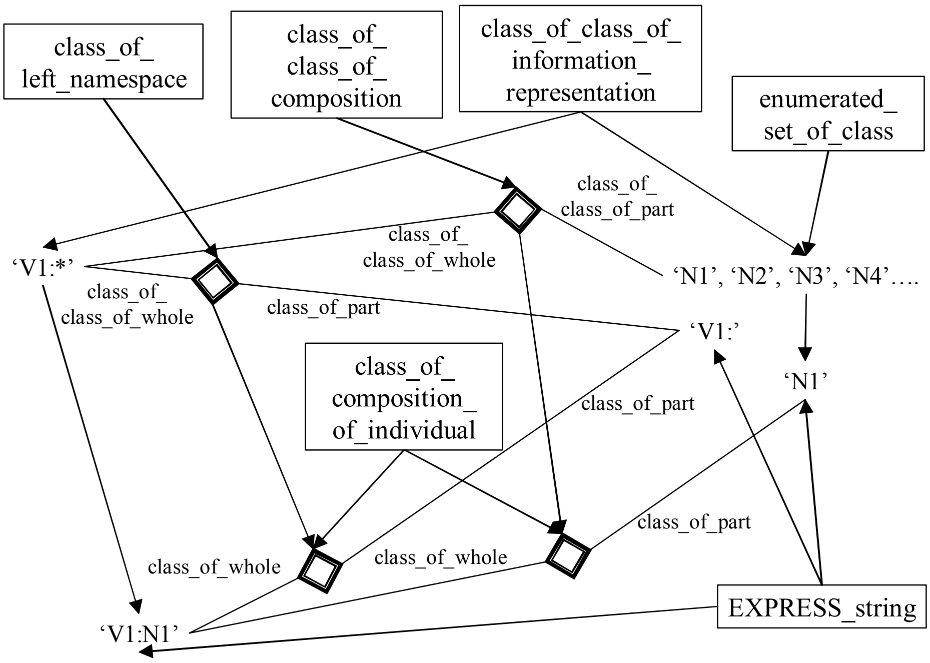


Figure 102 — Vessel V1 nozzle namespace

EXAMPLE 2 In Figure 103, the pattern ‘V1:*’ is used for identifiers of the nozzles of the vessel ‘V1’. A particular nozzle #4643 is identified by the string pattern ‘V1:N1’.

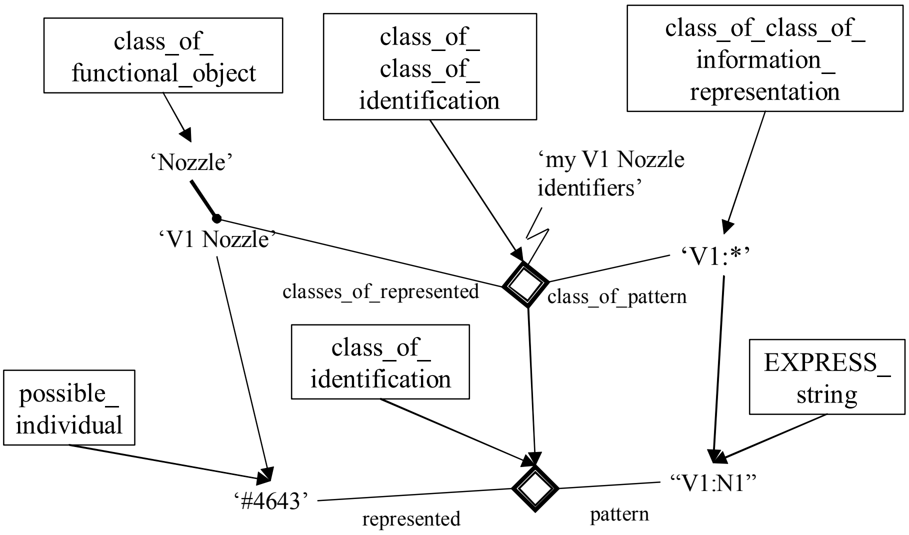


Figure 103 — Vessel V1 nozzle N1 identification

4.8.4.2.6 Uniform Resource Locators

In this part of ISO 15926, Uniform Resource Locator strings are handled as locational identifiers of particular productions of a document..

NOTE URLs are defined in IETF RFC 2396.

EXAMPLE In Figure 104, 'x' is a production of 'y', the final P&ID for Tag P101, stored at the URL 'http://www.designco.com/plant/pid/p101'. 'x' is a **functional_physical_object** as the specific materials (bits on a disc) of the production are not important.

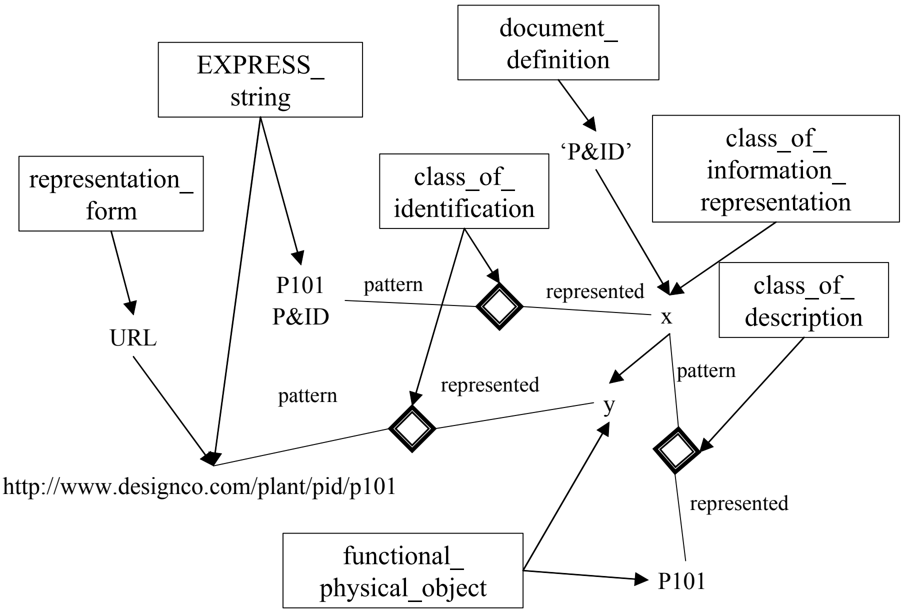


Figure 104 — P101 P&ID URL

4.8.4.3 Property

4.8.4.3.1 Property and class of property

A **property** is a **class_of_individual** whose member individuals have the same degree or magnitude of a quality or characteristic (see 5.2.26 and Figure 202). The types of quality or characteristic are defined using **class_of_property** (see 5.2.27 and Figure 203). **class_of_property** divides between property continuums and enumerated sets of properties. The property entity types are shown in Figure 105.

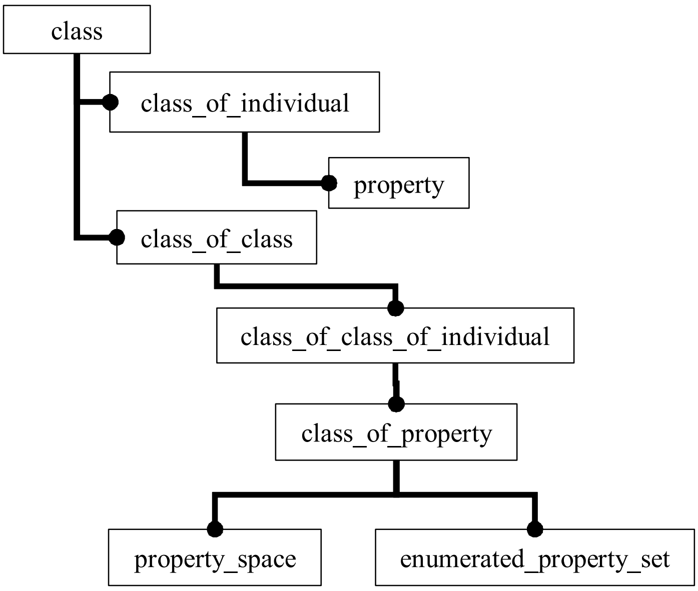


Figure 105 — Property and class of property

EXAMPLE The degree of hotness known as '21 C' is a **property** as shown in Figure 106. The quality of hotness is the **class_of_property** known as 'temperature'. The **possible_individual**s 'A' and 'B' both have 'temperature' of '21 C'.

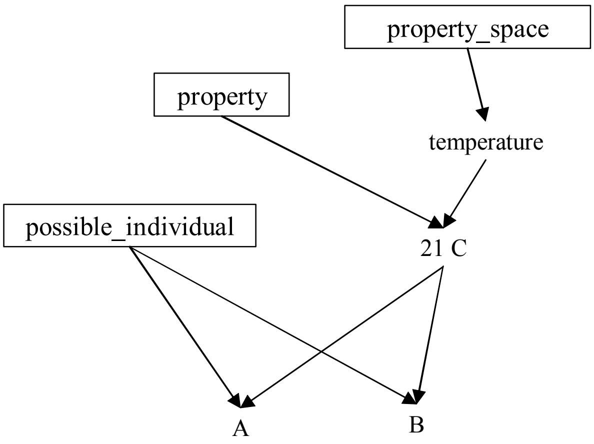


Figure 106 — Temperature property

4.8.4.3.2 Property quantification

A **property** is quantified by a **functional_mapping** relationship to particular number (see 5.2.26.6 and Figure 202). The **scale** used for the mapping is a **class_of_isomorphic_functional_mapping** (see 5.2.15 and Figure 191).

The model of **property_quantification** is shown in Figure 107. **property_quantification** is a **functional_mapping** relationship, indicating the **arithmetic_number** into which the **property** is mapped. **arithmetic_number** includes both the integer and real numbers and is described in 4.8.5.1 (see also 5.2.5.1 and Figure 181).

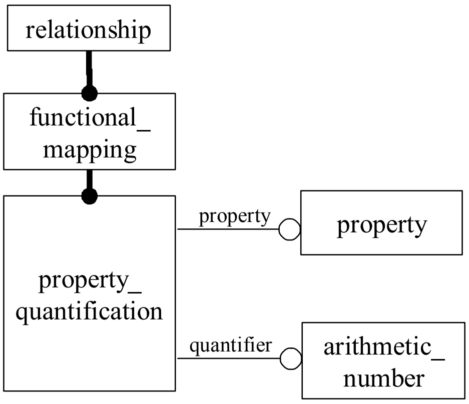


Figure 107 — Property quantification

property_quantification relationships may be classified by **scale** to indicate the units of the quantification**. scale** is a **class_of_functional_mapping**, whose members map members of the **property_space** domain to members of the codomain **number_space** as shown in Figure 108 (see 5.2.28 and Figure 204). A **property_space** is a continuum of properties (see 5.2.27.7 and Figure 203). A **number_space** is a continuum of numbers (see 5.2.5.10 and Figure 181). **real_number** and **integer_number** are **number_spaces**.

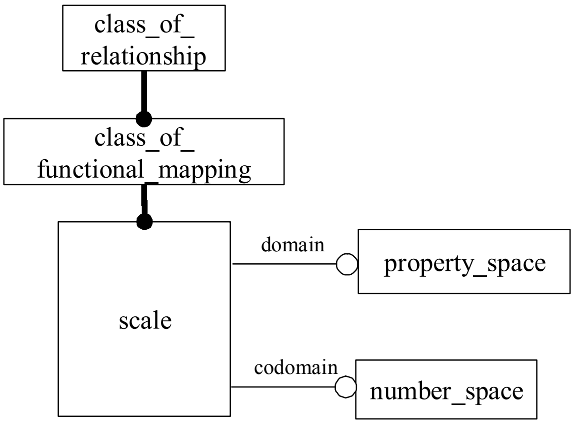


Figure 108 — Scale

EXAMPLE The temperature known as ‘21.0 C’ maps to the number 21.0 on the ‘Celsius’ scale as shown in Figure 109.

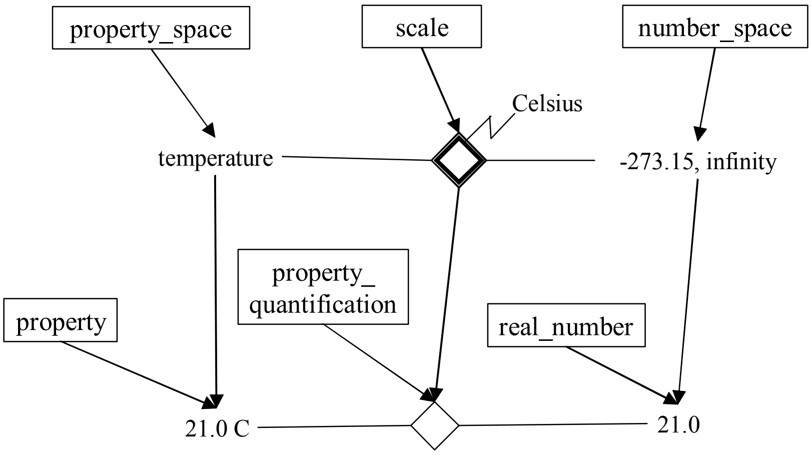


Figure 109 — Quantification of temperature 21.0 deg C

4.8.4.3.3 Indirect Property

indirect_property is relationship linking a **possible_individual** with a **property** (see 5.2.26.3 and Figure 202). The nature of the **indirect_property** is given by the **class_of_indirect_property** as shown in Figure 110 (see 5.2.26.1).

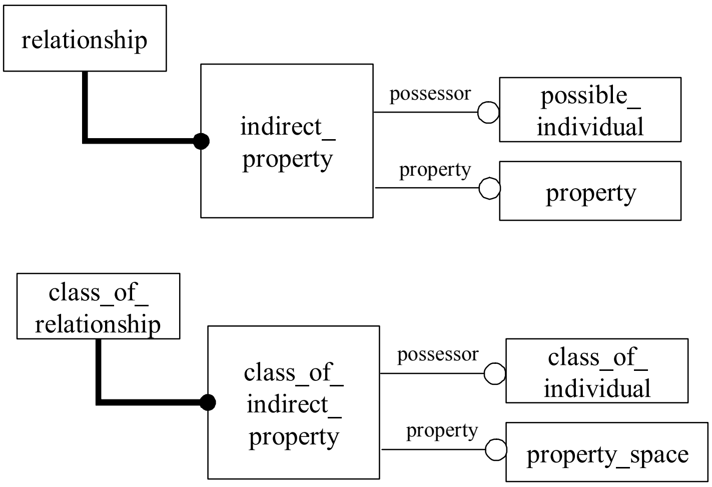


Figure 110 — Model of indirect property

In this part of ISO 15926, **indirect_property** shall be used when simple classification by a property is not possible. This arises when the property is derived or deemed from the properties of the **possible_individual** or from properties of particular parts of the **possible_individual** .

EXAMPLE 1 ‘pressure drop’ is a **class_of_indirect_property**. ‘pressure drop’ is not a **class_of_property** as its membership is indistinguishable from the membership of ‘pressure’. Figure 111 shows the instance data for a ‘Choke valve’ giving a ‘pressure drop’ of ‘10 bar’. #s1 is an operating temporal part of the ‘Choke valve’.

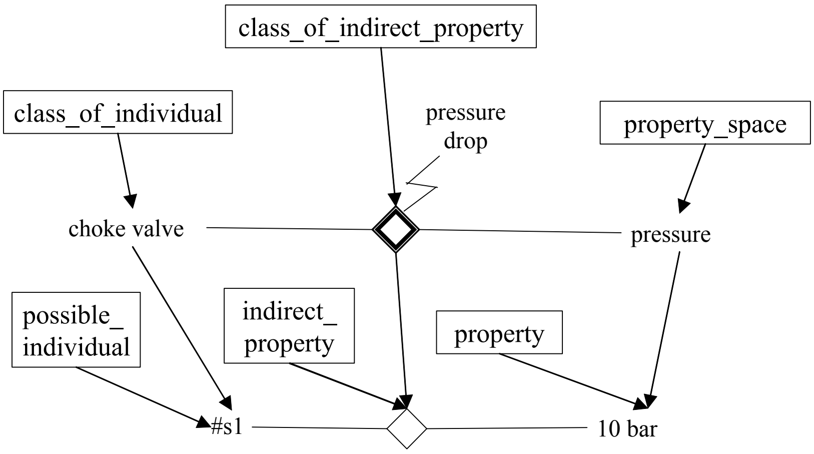


Figure 111 — Choke valve pressure drop

EXAMPLE 2 In Figure 112, a ‘container’ #C1 is rated to operate with an internal pressures less than 27 bar. The ‘Maximum Allowable Working Pressure’ is an **indirect_property**, referencing the ‘27 bar’ ‘pressure’ **property**. ‘Maximum Allowable Working Pressure’ is defined as a **class_of_indirect_property**.

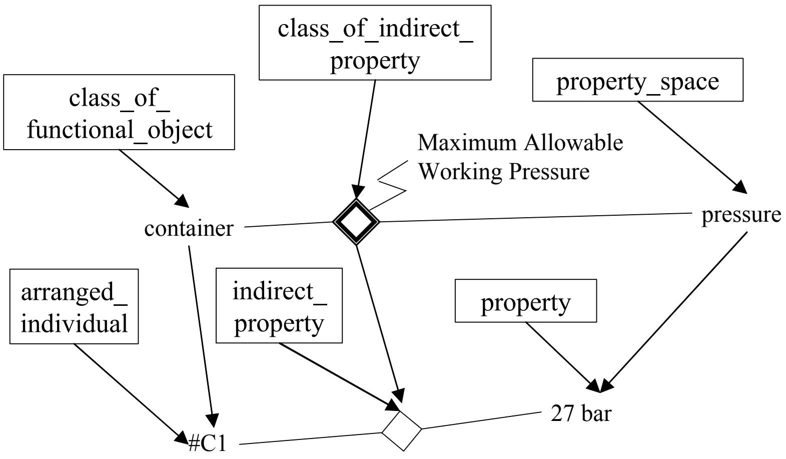

class_of_indirect_ property
class_of_ functional_object
property_space
Maximum Allowable Working Pressure
container
pressure
arranged_ individual
indirect_ property
property
27 bar
#C1

Figure 112 — Maximum allowable working pressure

4.8.4.3.4 Comparison of property

comparison_of_property is a **relationship** that indicates the order of two properties with respect to magnitude as shown in Figure 113 (see 5.2.26.2 and Figure 202).

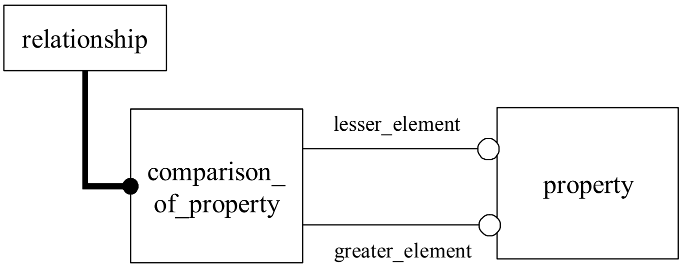


Figure 113 — Property comparison

EXAMPLE Figure 114 shows **property** ‘A’ is greater than **property** ‘B’.

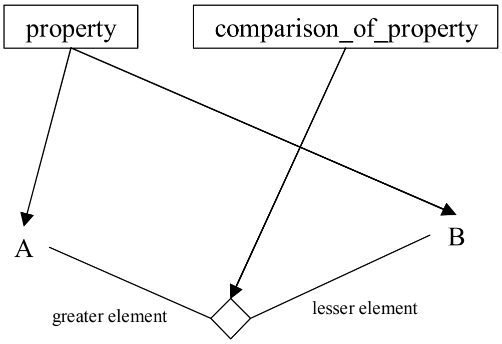


Figure 114 — Comparison of two properties

4.8.4.3.5 One-dimensional property space

A **property_space** is a type of **class_of_property** with a continuum of member properties. In this part of ISO 15926, two subtypes concerned with one-dimensional properties are defined as shown in Figure 115 (see also Figure 203).. A **single_property_dimension** is a single complete property continuum (see 5.2.27.8). A **property_range** is also a single continuum but restricted to the continuum between an upper and lower bound (see 5.2.27.6).

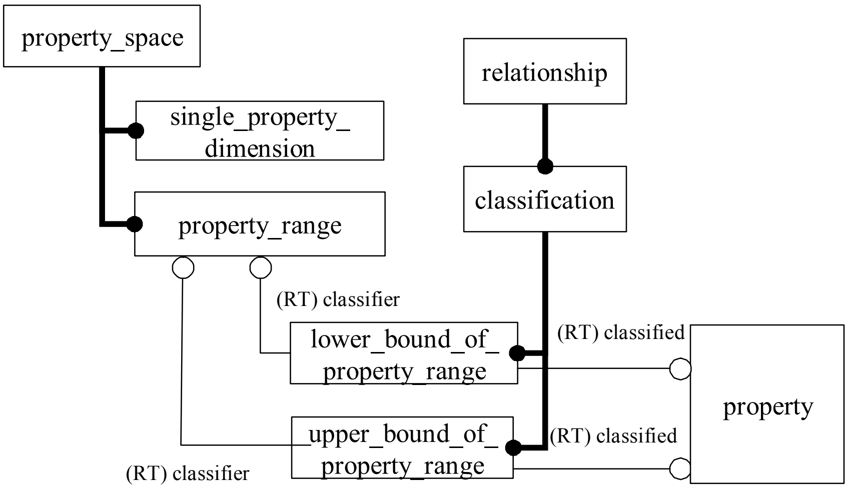


Figure 115 — Property spaces

EXAMPLE 1 ‘temperature’ is a **single_property_dimension**. The range of temperatures between ‘20 C’ and ‘40 C’ is a **property_range** as shown in Figure 116.

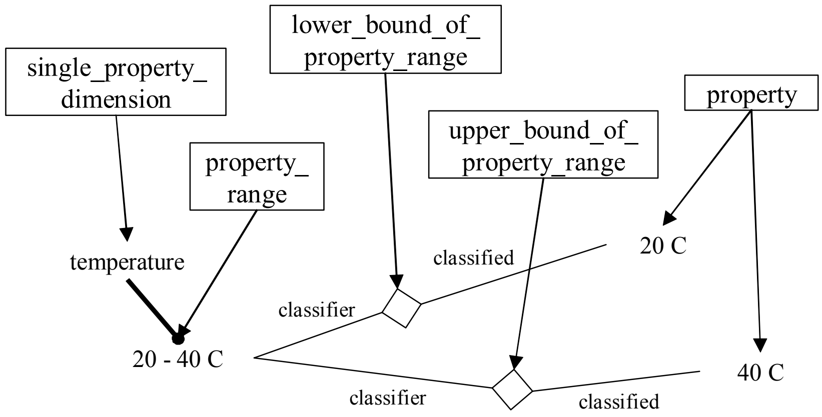


Figure 116 — Temperature range 20 – 40 C

property_ranges can be used to specify rules for membership of classes of individual, constraining the properties of the members.

EXAMPLE 2 A 'Type A Seal' has a 'working temperature' in the range of '20 to 40 C'. #S1, a temporal part of a particular seal, has a 'working temperature' of '24 C' as shown in Figure 117.

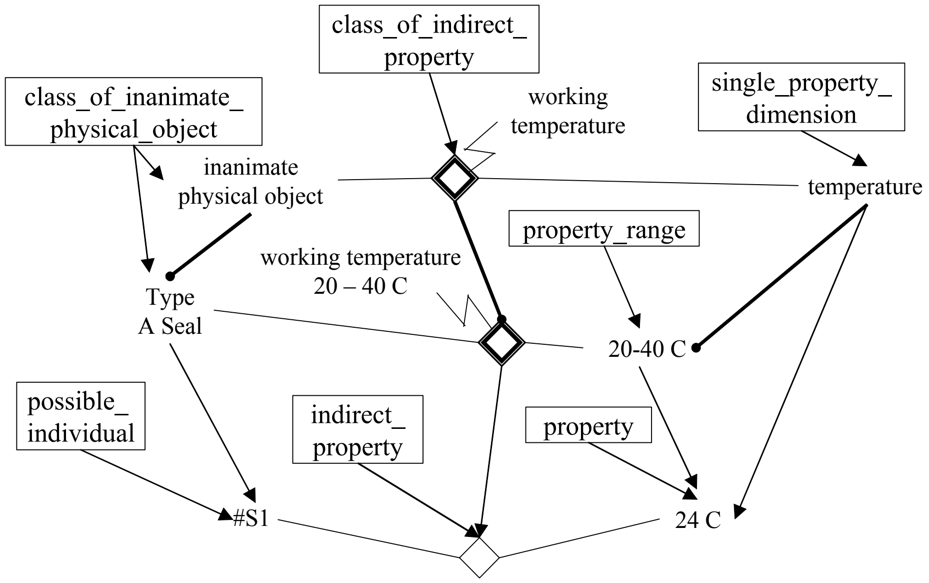


Figure 117 — Type A seal working temperature range

4.8.4.3.6 Multidimensional properties

A **multidimensional_property** is an ordered list of properties. A **multidimensional_property_space** classifies instances of **multidimensional_property** as shown in Figure 118 (see 5.2.27.5 and Figure 203).

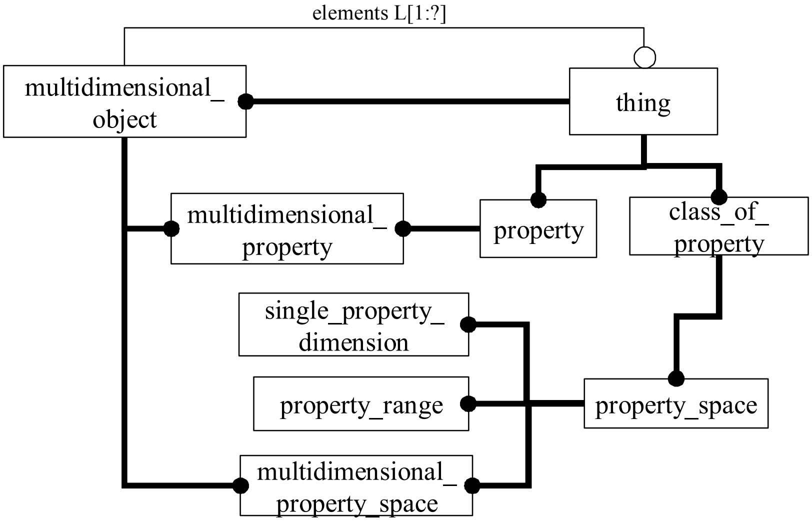


Figure 118 — Multidimensional property model

EXAMPLE 1 A pump flow head characteristic is a **multidimensional_object**. It consists of a continuum of Q, H property pairs, where Q is the flow rate and H is the flowing head difference. Each pair of properties Q_a and H_a, where Q_a is a particular flow rate and H_a is a particular head, is a **multidimensional_property** [$Q_{a,}$ $H_{a,}$]. The continuum of pairs of Q and H properties, shown as [Q,H], is a **multidimensional_property_space**. A particular set of pairs of Q and H, shown as $[Q,H]_l$ in Figure 119, corresponding to a particular pump head curve, is also a **multidimensional_property_space** [Q,H].

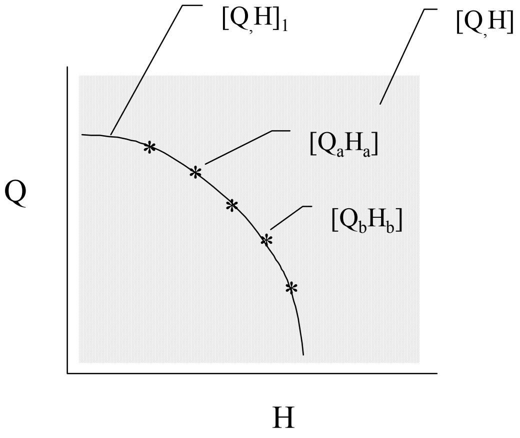


Figure 119 — A pump flow characteristic

EXAMPLE 2 The **multidimensional_property**s [$Q_{a,}$ $H_{a,}$] and [$Q_{b,}$ $H_{b,}$] are constructed as ordered pairs of properties. They are members of the **multidimensional_property_space**$[Q,H]_l$, which is a subset or specialization of the **multidimensional_property_space** [Q,H].In turn [Q,H] is constructed as the ordered pair of **single_dimensional_property**s Q and H as shown in Figure 120.

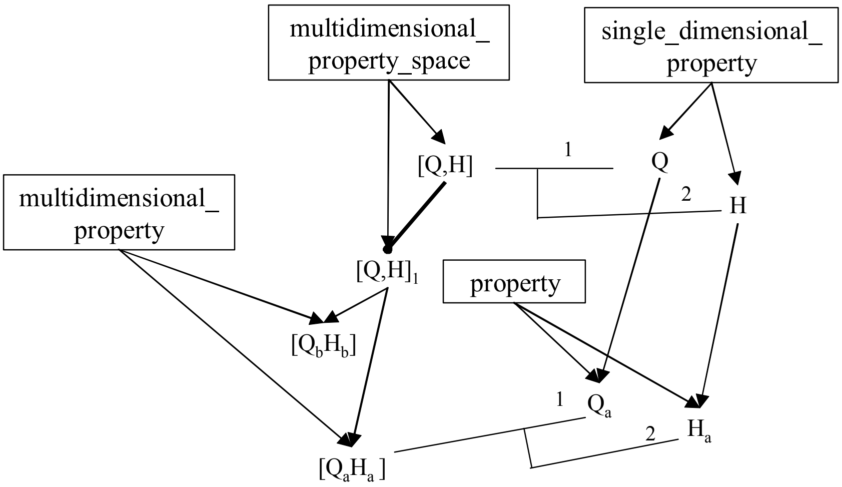

multidimensional_ property_space
single_dimensional_ property
[Q,H]
1
Q
2
H
multidimensional_ property
[Q,H]1
property
[QbHb]
1
Qa
2
Ha
[QaHa]

Figure 120 — Pump flow characteristic $[Q,H]_1$

EXAMPLE 3 Figure 121 shows a temporal part of an operating 'Type 24 Pump' #pz87 that has a flow rate of Q_a and a head of H_a. 'Type 24 Pump's have an operating characteristic of $[Q,H]_1$, meaning that any operating 'Type 24 Pump' has a QH that is a member of $[Q,H]_1$.

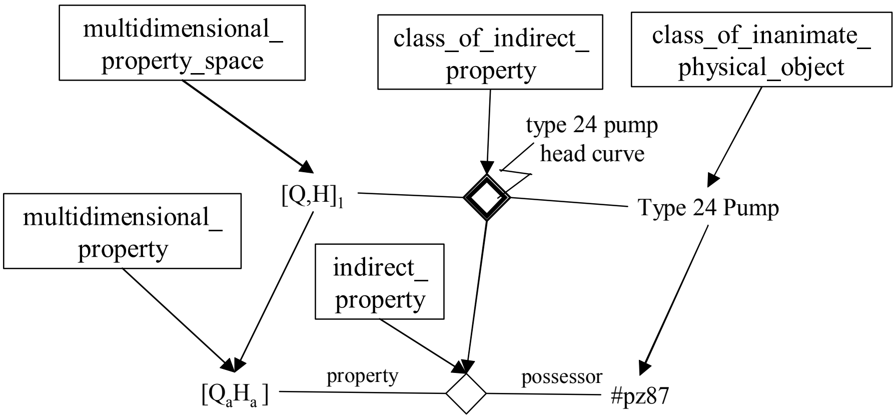

multidimensional_ property_space
class_of_indirect_ property
class_of_inanimate_ physical_object
type 24 pump head curve
[Q,H]1
Type 24 Pump
multidimensional_ property
indirect_ property
[QaHa]
property
possessor
#pz87

Figure 121 — Type 24 pump head flow characteristic

4.8.4.3.7 Position and coordinates

Position is also a **multidimensional_property**. In general, position in three-dimensional space can be defined by reference to three points. The three references form the dimensions of the **multidimensional_property**.

The position properties can be mapped to **multidimensional_number**s. A particular position can have several mappings, corresponding to different **coordinate_system**s.

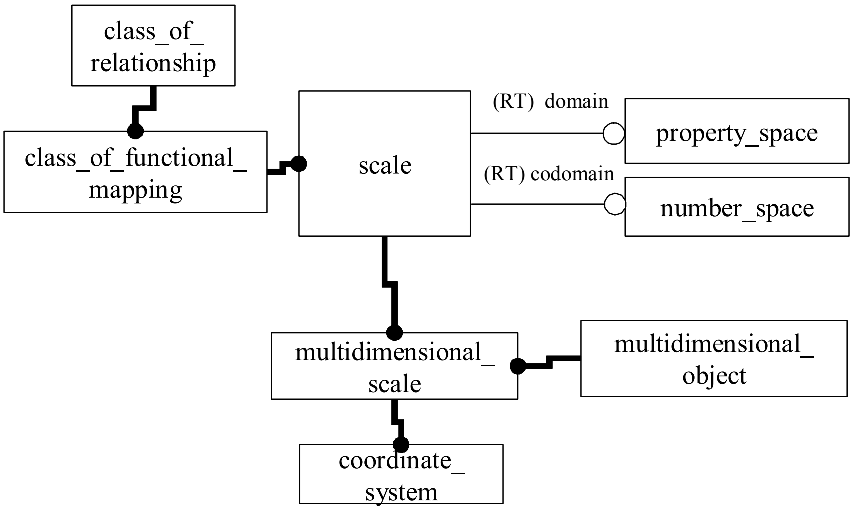


Figure 122 — Coordinate system model

coordinate_system is a **multidimensional_scale** that maps a **property_space** to a **multidimensional_number_space** (see 5.2.28.2 and Figure 204). The model is shown in Figure 122. The **coordinate_system** in a three-dimensional space is defined by three of its members.

EXAMPLE Figure 123 shows where the **possible_individual** 'A' has the position described as [2,3,4] in **coordinate_system** known as ‘CS 21’.

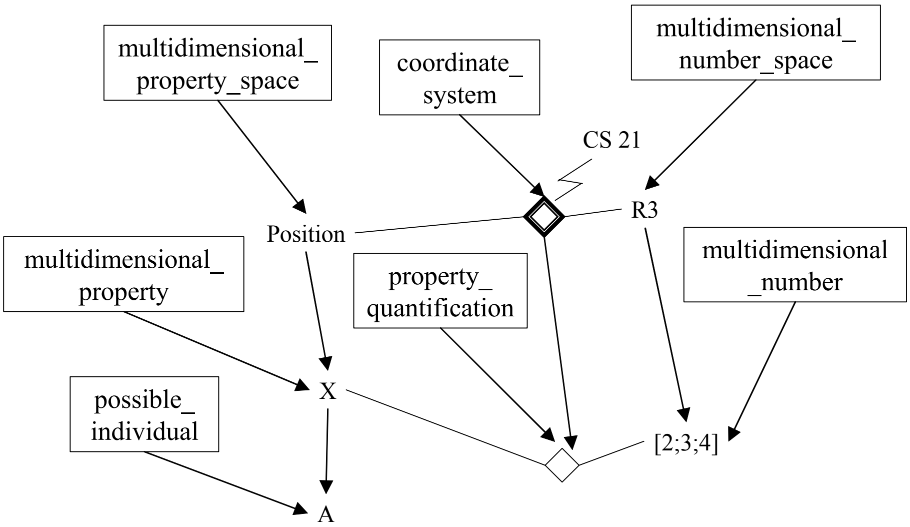


Figure 123 — Coordinate system CS 21

4.8.4.4 Status and class of status

possible_individual may be classified by **status** (see 5.2.7.13 and Figure 183). A **status** reflects the condition or state of **possible_individual**s that are not quantifiable and have no ordering.

A **status** is distinguished from a **property** by being generally descriptive and not quantified by mappings to numbers where the order is significant. Different types of **status**es may be distinguished as **class_of_status** (see 5.2.7.11). The model diagram for this is shown in Figure 124.

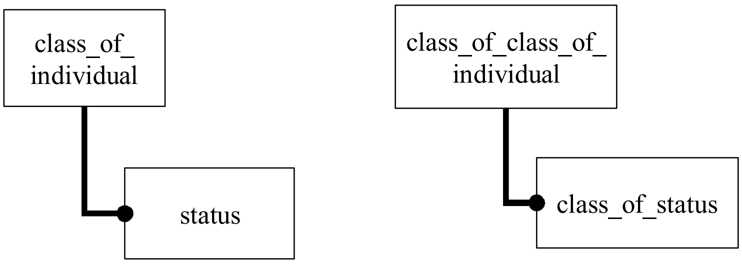


Figure 124 — Status and class of status

EXAMPLE 1 'pitted', 'gouged', 'scratched', and 'dented' are examples of **status** that are members of the **class_of_status** 'surface condition'. Figure 125 shows the data for this. The 'pipe exterior' #rc359 is an **actual_individual** classified by the 'dented' **status**.

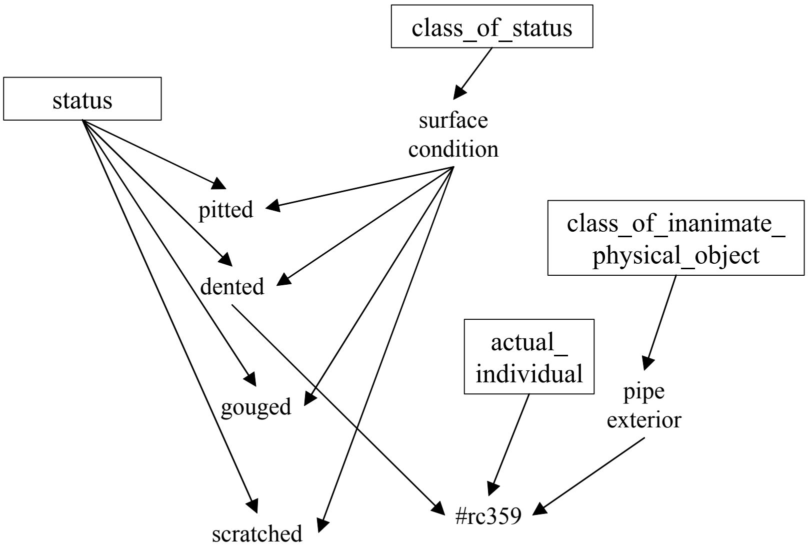


Figure 125 — Surface condition statuses

EXAMPLE 2 A 'degree of wear', such as 'no wear', 'slightly worn', 'very worn', and 'worn out', is not a **status** as it can be ordered and mapped to a number scale.

EXAMPLE 3 A 'degree of openness of a valve' is not a **status** as it can be ordered and mapped to a number scale.

4.8.4.5 Shape and dimension

4.8.4.5.1 Individual dimension

Individuals that show spatial symmetry are often described using dimensions. In this sense, a dimension is a **class_of_individual**, where the members have a common length and various intersections with boundaries of the individual they are dimensions of (see 5.2.29.6 and Figure 205). The model of **individual_dimension**s is shown in Figure 126.

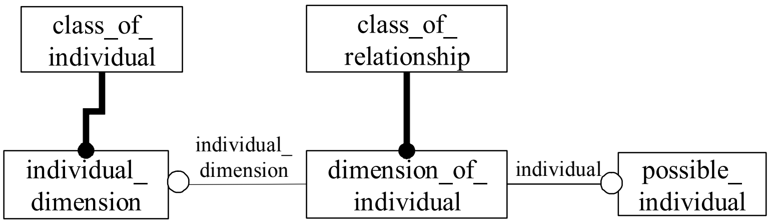


Figure 126 — Dimension of individual

dimension_of_individual is a **class_of_relationship** whose members indicate that an **individual_dimension** is a dimension of a **possible_individual** (see 5.2.29.4).

EXAMPLE 1 The width of 'table' is an **individual_dimension** as shown in Figure 127.

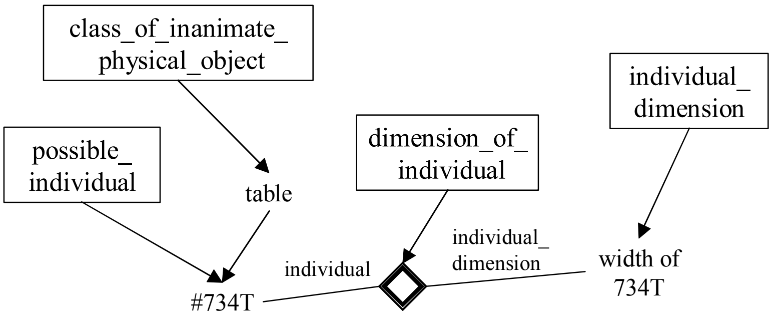


Figure 127 — Width of my table

An **individual_dimension** is not necessarily part of the individual for which it is a dimension.

EXAMPLE 2 The internal diameter of a pipe is not part of the pipe.

4.8.4.5.2 Property of individual dimension

Members of an **individual_dimension** often have one or more common properties. A type of **specialization** relationship is defined to allow this to be stated as shown in Figure 128 (see 5.2.29.11 and Figure 205).

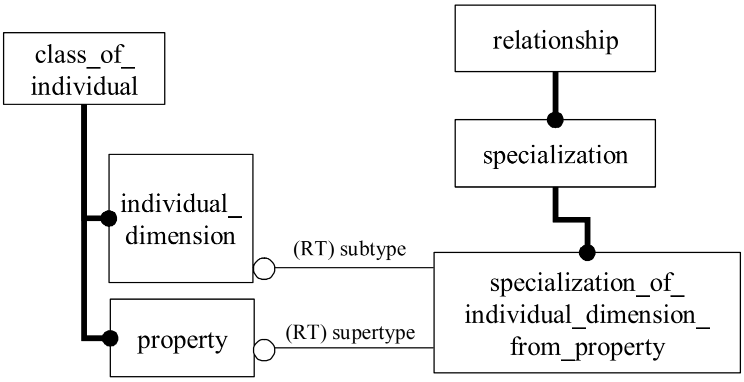


Figure 128 — Properties of individual dimension

EXAMPLE Figure 129 shows the table width has a length of exactly '520 mm'.

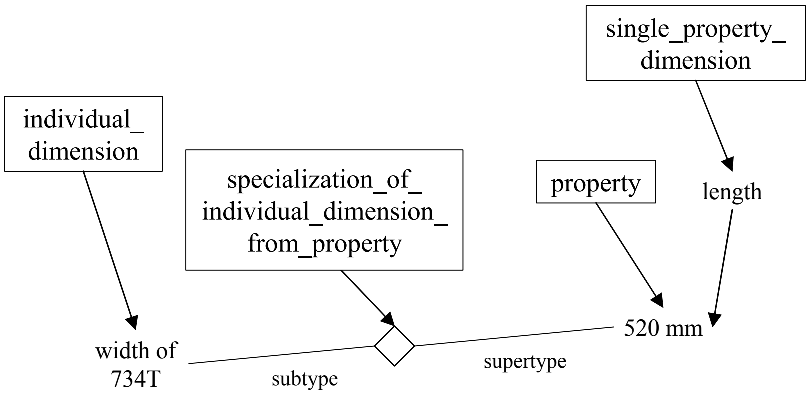


Figure 129 — My table width of 520 mm

4.8.4.5.3 Shape

A **shape** describes the constant nature of relative spatial positions of a boundary (see 5.2.29 and Figure 205). **shape** is a **property** and a **class_of_individual** as shown in Figure 130.

class_of_shape enables different types of **shape** such as circles, rectangles to be recognised.

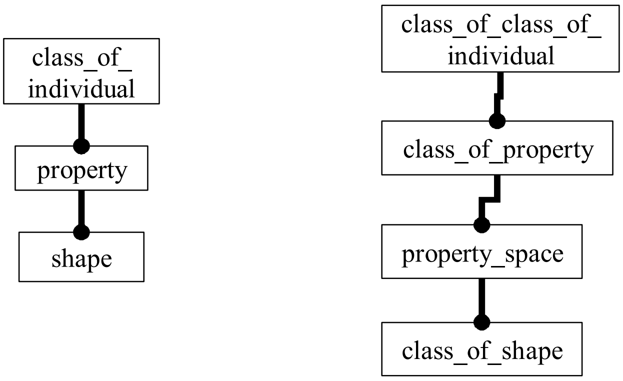


Figure 130 — Shape and class of shape

EXAMPLE Figure 131 shows that a **possible_individual** that measures 2x1 cm is a ‘2x1 cm’ **shape** that is classified as a ‘Rectangle’.

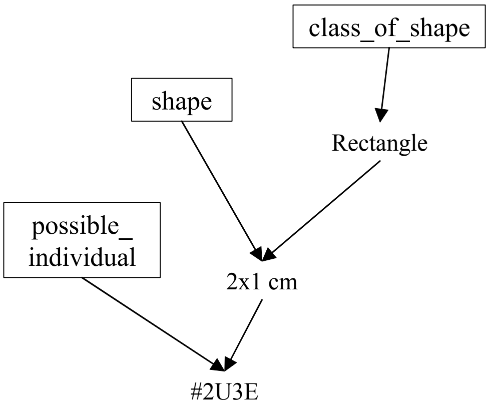


Figure 131 — Rectangle shapes

4.8.4.5.4 Dimension of shape

shapes may also have dimensions that are common to all members of the **shape**. A **shape_dimension** is a **class_of_class_of_individual**, where the members have a common length and common intersections with the boundaries of the **shape** members they are dimensions of (see 5.2.29.10). The types of intersections are distinguished by **class_of_shape_dimension** (see 5.2.29.3). The model for this is shown in Figure 132.

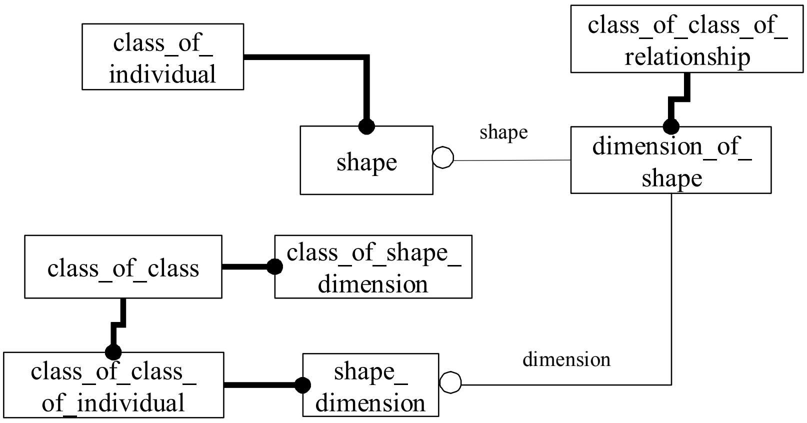


Figure 132 — Dimension of shape

dimension_of_shape is a **class_of_class_of_relationship**, whose members link a member of the **shape** class with a member of the **dimension_of_shape** class (see 5.2.29.5). A **dimension_of_shape** can classify a member of **dimension_of_individual** (see 5.2.29.4).

EXAMPLE Figure 133 shows where '10cm circle's are **shape**s. A '10cm diameter' is a **shape_dimension** of '10cm circle's. A member of '10cm diameter' is a dimension of member of '10cm circle'.

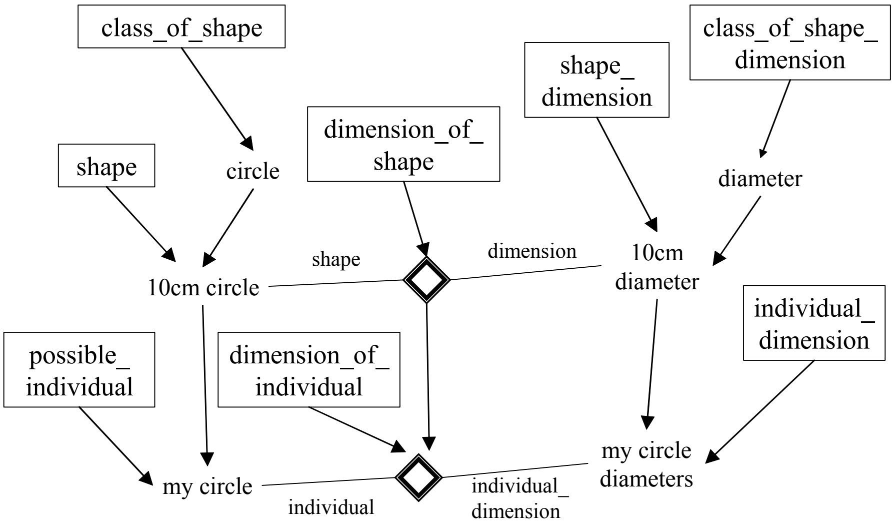


Figure 133 — 10cm diameter circle

4.8.4.5.5 Property for shape dimension

Members of a **shape_dimension** have one or more common properties. The **class_of_relationship property_for_shape_dimension** is defined to allow this to be stated as shown in Figure 134 (see 5.2.29.7 and Figure 205).

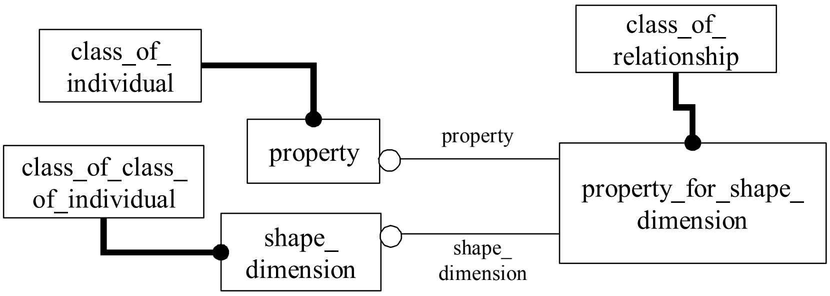


Figure 134 — Property for shape dimension

EXAMPLE Figure 135 shows the members of **shape_dimension** "10cm diameter" all have the 'length' of '10 cm'.

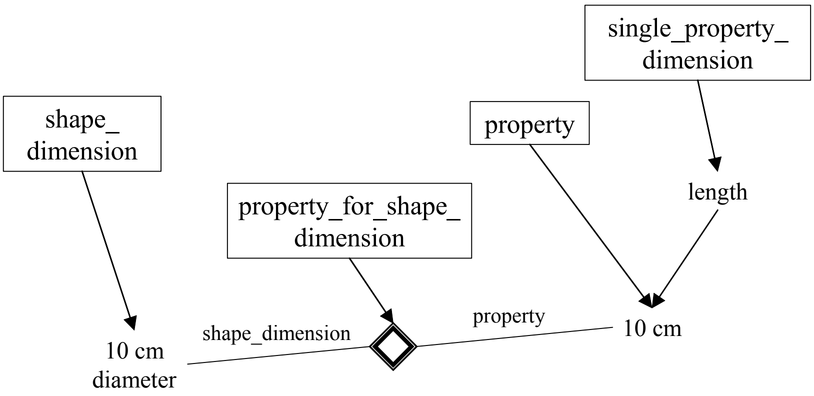


Figure 135 — Diameters of 10 cm length

4.8.4.5.6 Class of shape dimension

shape_dimensions can be classified by how the dimension is positioned with respect to the individual of which it is a dimension. **class_of_shape_dimension** is defined for this purpose as shown in Figure 136 (see 5.2.29.3 and Figure 205).

EXAMPLE 1 'diameter', 'height', 'length', and 'width' are **class_of_dimension_for_shape**.

class_of_dimension_for_shape enables a type of dimension to be characteristics of a type of **shape**.

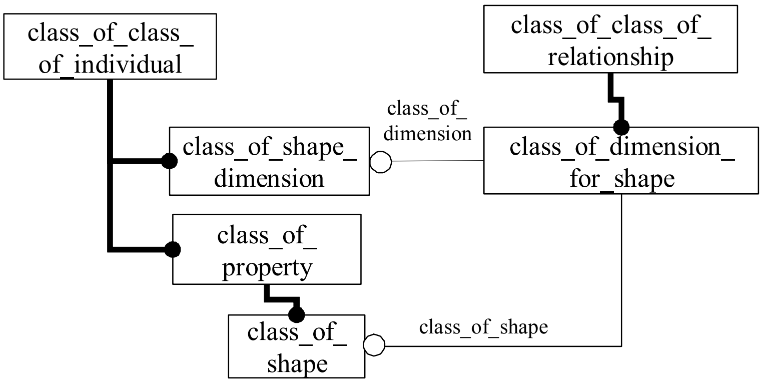


Figure 136 — Class of shape dimension

EXAMPLE 2 A ‘diameter’ is a dimension of a ‘circle’ as shown in Figure 137.

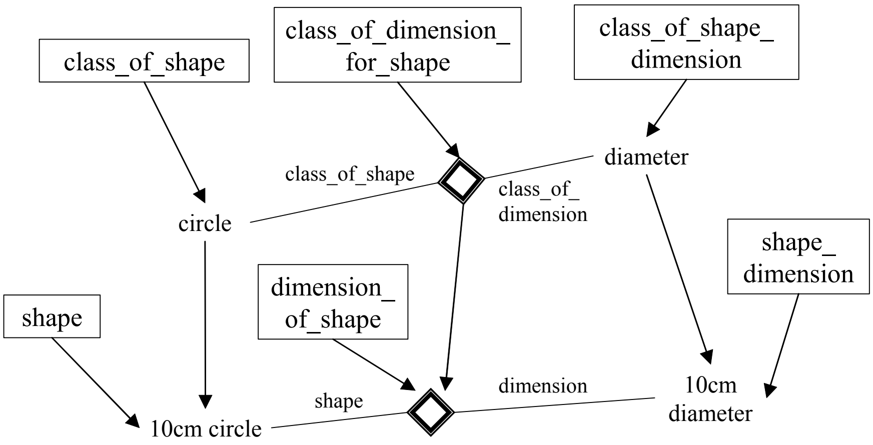


Figure 137 — Diameters of circles

4.8.4.5.7 Class of property for class of shape dimension

Certain types of **shape_dimension** have certain types of **property** (see 5.2.29.8 and Figure 205). The model diagram supporting this is shown in Figure 138.

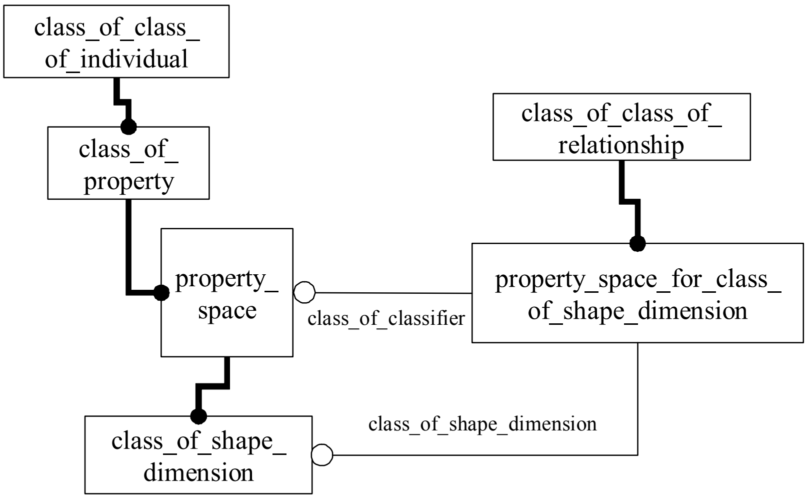

class_of_class_ of_individual
class_of_ property
property_ space
class_of_class_of_ relationship
property_space_for_class_ of_shape_dimension
class_of_classifier
class_of_shape_ dimension
class_of_shape_dimension

Figure 138 — Shape dimension property classes

EXAMPLE That a ‘diameter’ has a ‘length’ is shown in Figure 139.

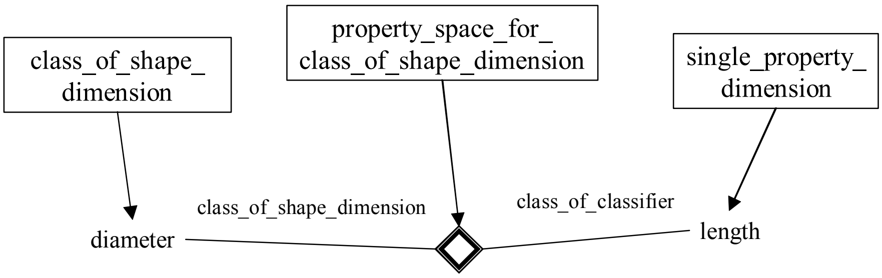

class_of_shape_ dimension
property_space_for_ class_of_shape_dimension
single_property_ dimension
class_of_shape_dimension
class_of_classifier
diameter
length

Figure 139 — Diameter lengths

4.8.4.6 Class of event and point in time

An **event** and a **point_in_time** may be classified. **event** classes may be determined by the **class_of_point_in_time** of the member **event**s (see 5.2.7.10 and Figure 183). The model for this is shown Figure 140.

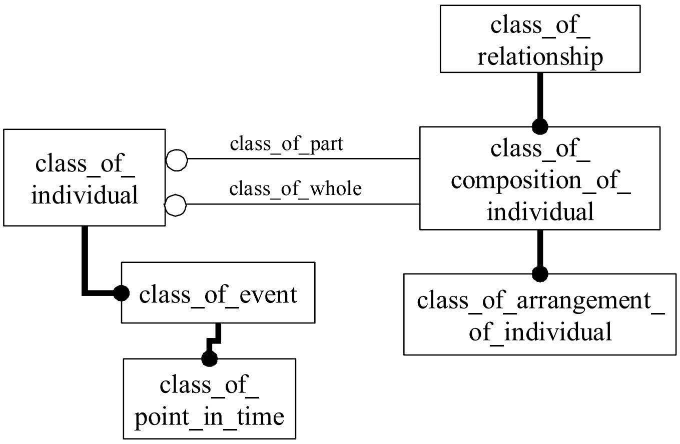


Figure 140 — Class of event and point in time

EXAMPLE Figure 141 shows 'take off' is a **class_of_event** that marks the transition of an aircraft from being in contact with land to being airborne. 'midnight' is a **class_of_point_in_time**. 'Midnight take off' is a **class_of_event** that is a specialization of 'take off'. All 'Midnight take off's are part of a 'midnight' **point_in_time**.

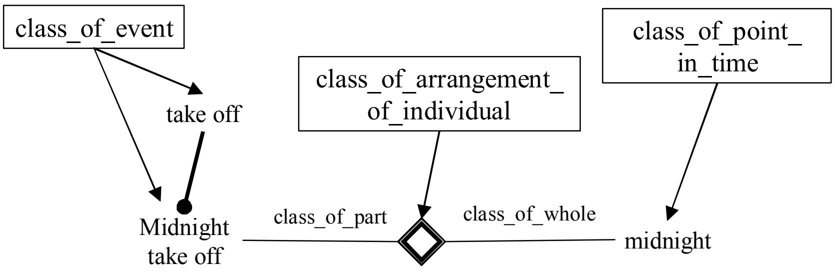


Figure 141 — Midnight takeoff events

4.8.4.7 Class of period in time

period_of_time may be classified (see 5.2.7.9 and Figure 183). **class_of_temporal_sequence** enable members of **period_of_time** classes to be ordered (see 5.2.22.3 and Figure 198). The model for this is shown in Figure 142.

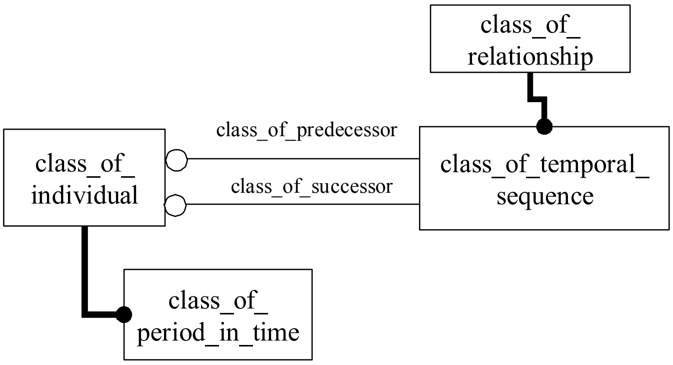


Figure 142 — Class of period in time

EXAMPLE Figure 143 shows 'June' and 'July' are **class_of_period_in_time**. Each 'July' follows a 'June'.

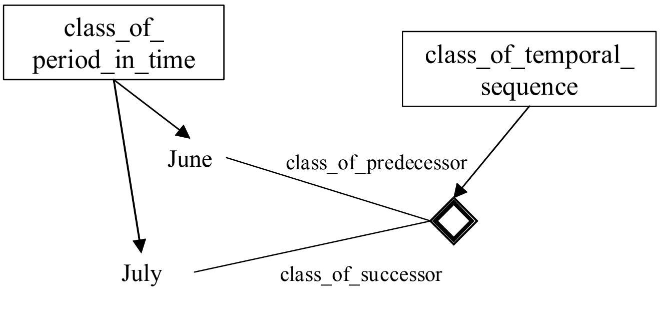


Figure 143 — July follows June

4.8.4.8 Role and domain

role_and_domain is a type of **class** based on what something does in the context of an **activity** or a **relationship** (see 5.2.13.5 and Figure 189). Some **role_and_domain** classes are purely **role** only, that is their membership is unconstrained by any other consideration apart from **role** (see 5.2.13.4). The subtype of **role** is defined for these. However, most **role_and_domain** classes are intersections of a pure **role** with a domain that restricts the type of thing that can act in the **role**. The model for this is shown in Figure 144.

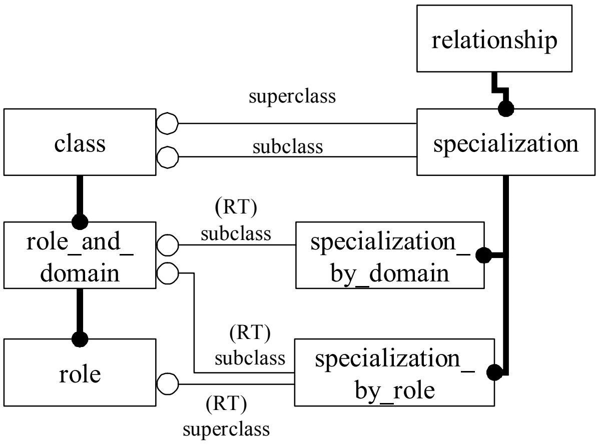


Figure 144 — Role and domain

The specialization relationship **specialization_by_role** is defined to enable the **role** superclass of a **role_and_domain** to be defined (see 5.2.13.7). The **specialization_by_domain** relationship enables the domain superclass of a **role_and_domain** to be defined (see 5.2.13.6).

EXAMPLE Figure 145 shows 'controller' is a **role**. 'Controller person' is **role_and_domain** that is a specialization of the **role** 'controller' and of the domain 'person'. The 'controller person' **role_and_domain** excludes mechanical 'controller's.

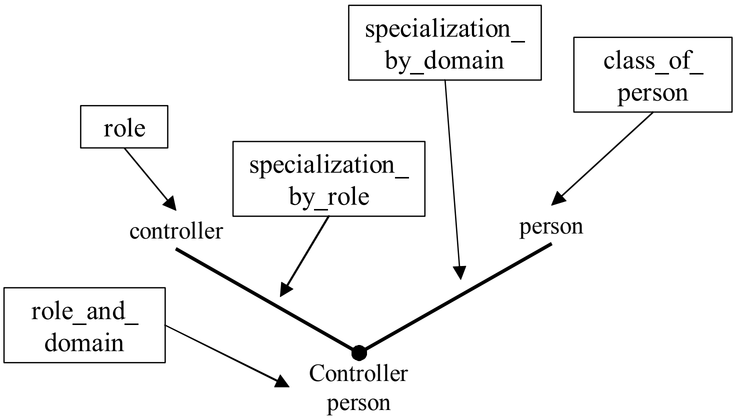


Figure 145 — Controller person role and domain

4.8.4.8.1 Intended and possible roles

Some individuals, by their characteristics and properties, are suited to play particular roles. This applies to individuals and to the members of particular classes of individuals (see 5.2.24 and Figure 200). The types of **relationship** and **class_of_relationship** of this nature recognized by the model are shown in Figure 146.

intended_role_and_domain relationships indicate the **role_and_domain**s that an individual is for, by design or otherwise (see 5.2.24.3).

EXAMPLE 1 A particular '2m steel bar with a tapered end' is intended to play the role of 'lever'.

A **possible_role_and_domain** relationship indicates a **role_and_domain** that an individual could undertake, not by design, but because the individual has properties that are suitable for the role (see 5.2.24.4).

EXAMPLE 2 A particular 'cube of concrete of mass 20 kg' could play the role of 'anchor'.

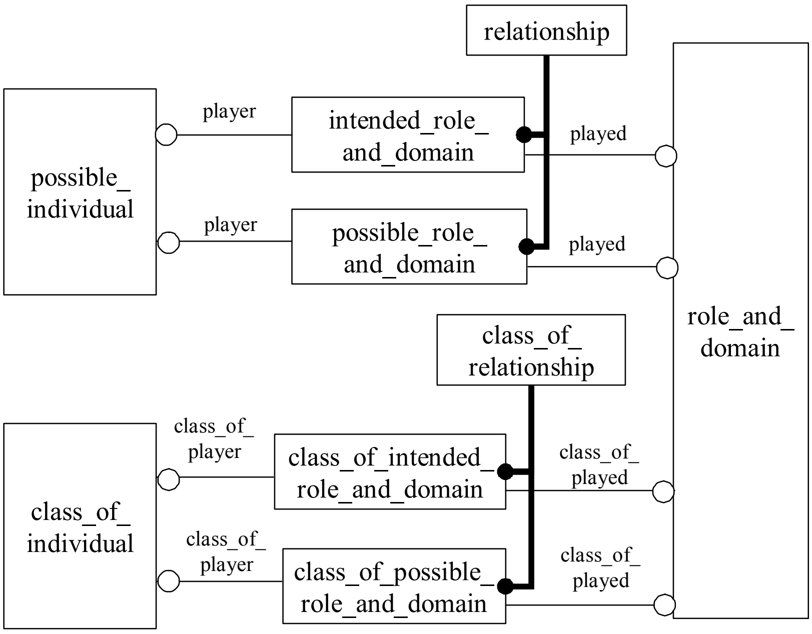


Figure 146 — Intended and possible role and domain

A **class_of_intended_role_and_domain** indicates a **role_and_domain** that members of the **class_of_individual** are designed to play (see 5.2.24.1). The characteristics and properties that are common to the members of the **class_of_individual** have been deliberately chosen to enable members to perform in the designated **role_and_domain**.

It is important to distinguish **role_and_domain** classes from **class_of_functional_object** classes. Members of **role_and_domain** classes are involved in an activity, whereas members of **class_of_functional_object** classes have the capability to be involved. Often these classes, though having different meanings, have the same names.

EXAMPLE 3 Figure 147 shows members of 'pump', the **class_of_functional_object**, are intended to play the role of 'pumper'. 'Pumper' is a specialization of the **role_and_domain** 'performer'. Things which are pumpers are actually pumping.

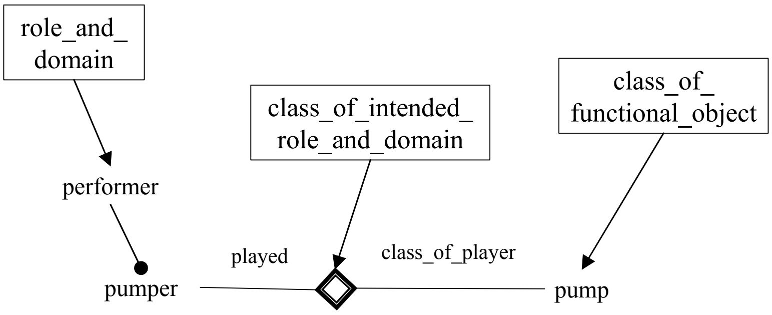


Figure 147 — Intended performer role for pumps

class_of_possible_role_and_domain indicates a **role_and_domain** that members of the **class_of_individual** can play though not by purpose of their design (see 5.2.24.2).

EXAMPLE 5 Members of 'Cup', the **class_of_functional_object**, can play the role of 'sugar bowl'.

4.8.4.8.2 Participating role and domain

In general any **thing** can be a member of a **role_and_domain** class, this includes individuals and abstract objects. However, because participants in activities must be individuals, the subtype **participating_role_and_domain**, which restricts members to being individuals, is defined (see 5.2.13.3 and Figure 189). The model for this is shown in Figure 148.

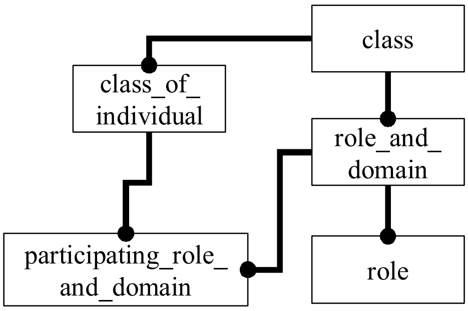


Figure 148 — Participating role and domain

4.8.4.9 Class of activity

Classes of activity can be defined by restricting the types of the things involved in the activity class members (see 5.2.10 and Figure 186). The model for this is shown in Figure 149.

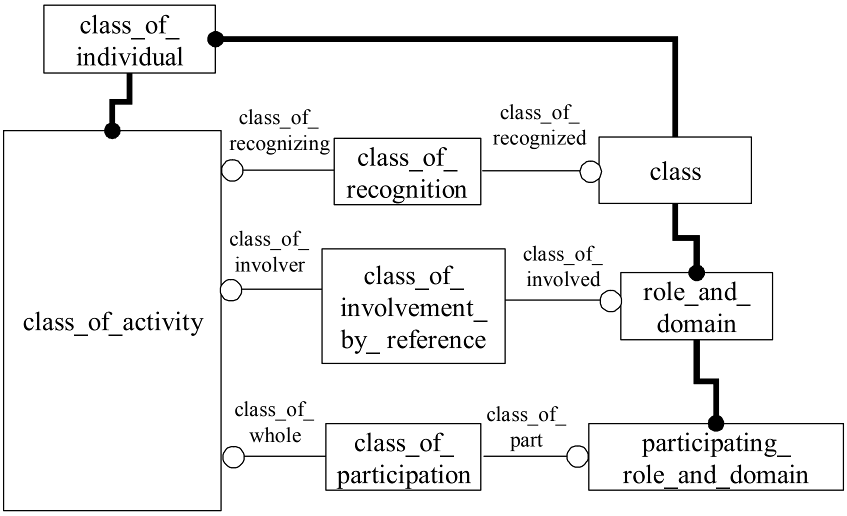


Figure 149 — Class of activity

EXAMPLE 1 'Fluid pressure measurement using an instrument type 167' is a **class_of_activity** and is a specialization of the 'measurement' **class_of_activity**.

class_of_participation constrains the types of individuals and their role in the activity class members (see 5.2.10.5). A **participating_role_and_domain** is the intersection of a **role_and_domain** and a **class_of_individual**.

EXAMPLE 2 Figure 150 shows the 'instrument type 167' used as a 'Measurer' is a **participating_role_and_domain**. A **class_of_participation** linking the fluid measurement class to this **participating_role_and_domain** constrains the measurement to be undertaken with a 'instrument type 167'. #789 is a temporal part of a 'type 167 instrument' that participates in the measuring **activity** #1234.

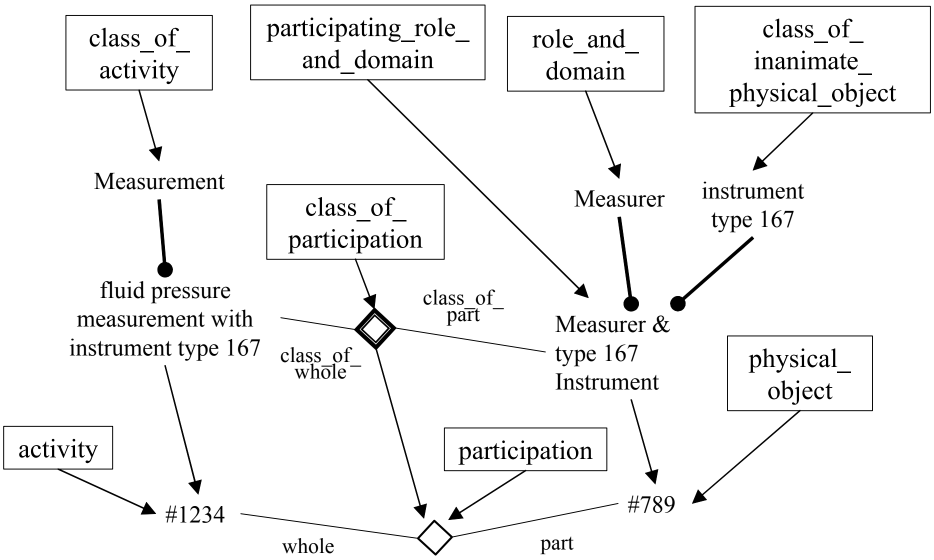


Figure 150 — Fluid measurement activity using type 167 instrument

Any number of **participating_role_and_domains** may be defined for a **class_of_activity**.

EXAMPLE 3 Figure 151 shows 'fluid' in the role of 'Measured' is a **participating_role_and_domain**. A **class_of_participation** linking the fluid measurement class to this **participating_role_and_domain** constrains the measurement to measure a fluid. #S27 is a temporal part of a **stream** that is measured by the measuring **activity** #1234.

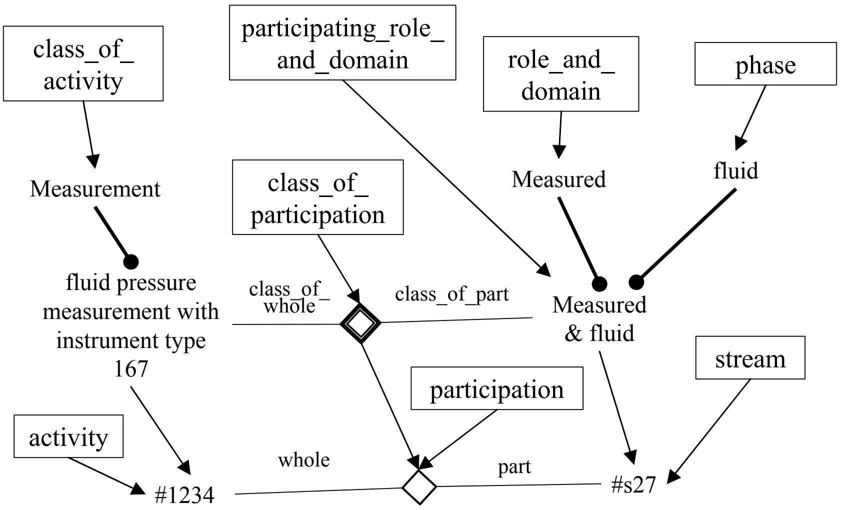


Figure 151 — Fluid pressure measurement activity

class_of_recognition is a **class_of_relationship** that indicates the **class** of the **thing**s that may be recognised as a result of a member of the **class_of_activity** (see 5.2.10.6).

EXAMPLE 4 Figure 152 shows a fluid measurement activity results in the recognition of a classification of the measured fluid to a 'pressure' property class. The particular measurement **activity** #1234 results in the **classification** of #S27 by the '26 bar' 'pressure' class.

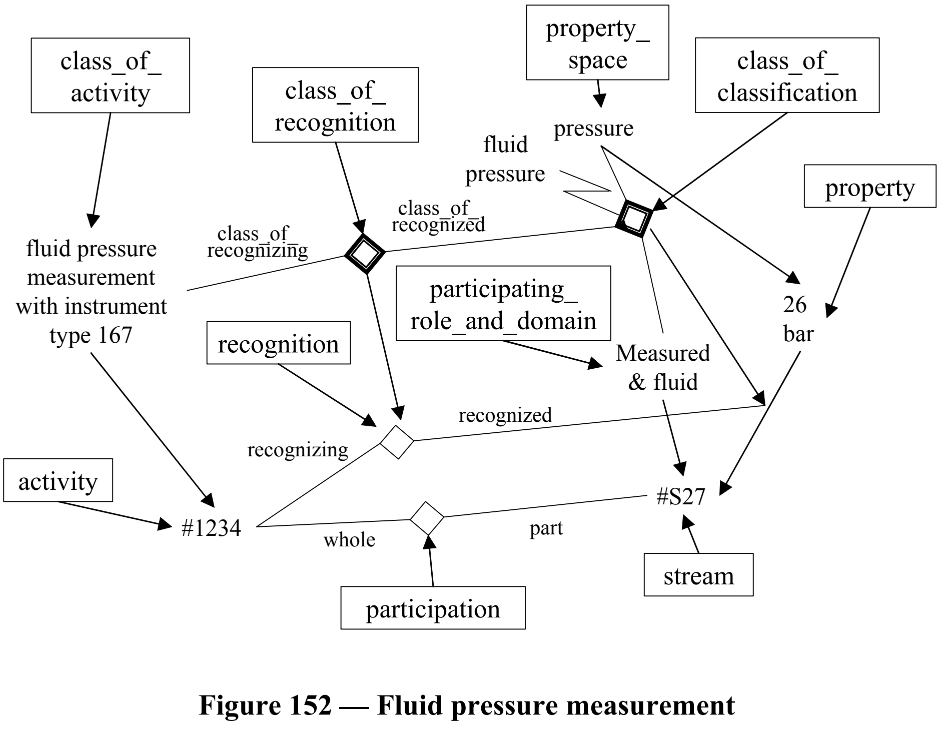


Figure 152 — Fluid pressure measurement

4.8.4.10 Class of class of individual

Other subtypes of **class_of_individual**, not explicitly modelled, can be defined as instances of **class_of_class_of_individual** and used to classify the required **class_of_individual** (see 5.2.7.4 and

Figure 183). Some important types of **class_of_class_of_individual** are explicitly modelled as shown in Figure 153.

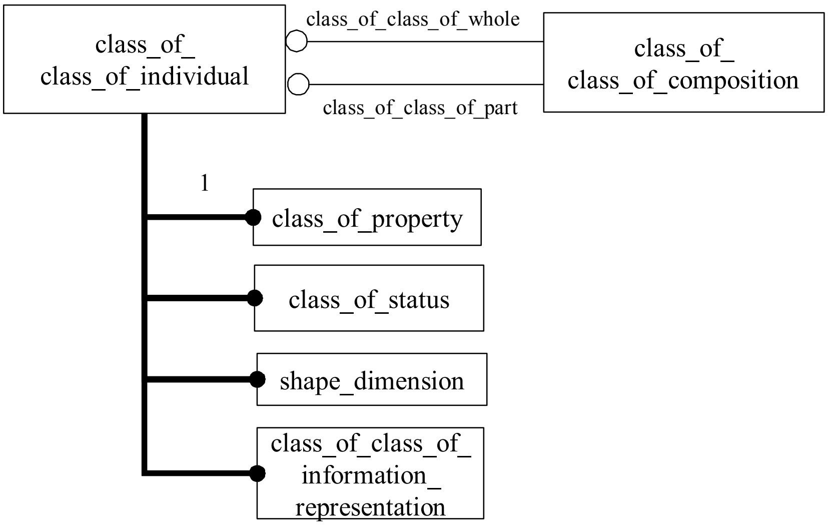


Figure 153 — Class of class of individual

4.8.5 Numbers

4.8.5.1 Arithmetic number

arithmetic_number is a **class_of_class** as shown in Figure 154 (see 5.2.5.1 and Figure 179). In this part of ISO 15926, integer numbers are distinguished from real numbers. Both are single-dimensional numbers.

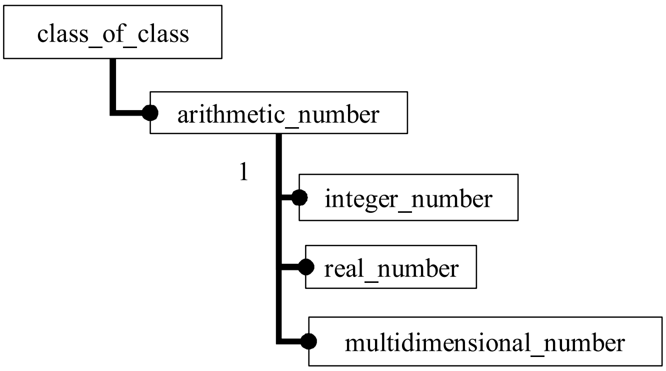


Figure 154 — Arithmetic number

NOTE Representations of numbers is intended to be handled with **class_of_information_representation**.

EXAMPLE 1 Figure 155 shows that there is a class ‘x’ that is a **real_number** represented by the **EXPRESS_real** 12.7.

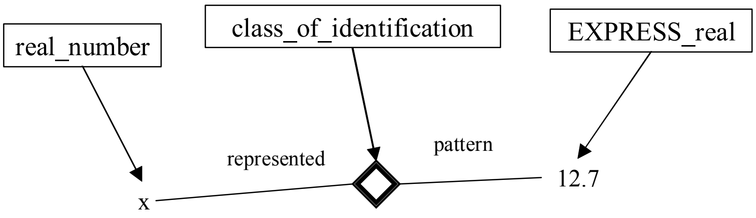


Figure 155 — Representation of real number

multidimensional_number enables ordered pairs, triples etc. of **arithmetic_number**s to be defined (see 5.2.5.7, Figure 180 and Figure 181).

EXAMPLE 2 Figure 156 shows the coordinate triple [1.2, 2.3, -6.8] is a **multidimensional_number**. The order, given by the element number, is significant. The triple [2.3, 1.2, -6.8] is a different triple.

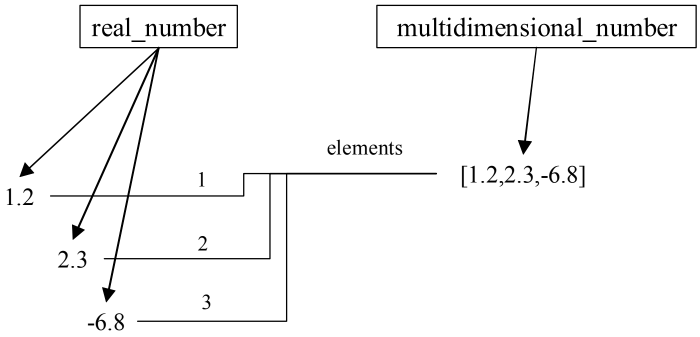


Figure 156 — Multidimensional number

4.8.5.2 Class of number

class_of_number is a **class_of_class** that includes both discrete and continuous sets of numbers (see 5.2.5.3, Figure 179 and Figure 181). The model is shown in Figure 157.

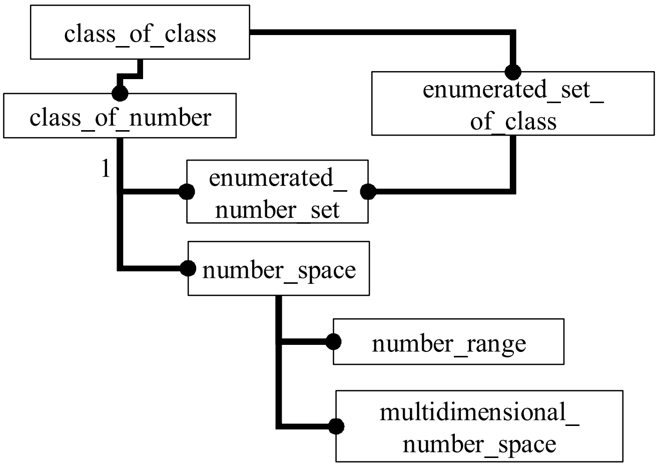


Figure 157 — Class of number

4.8.5.2.1 Enumerated number set

enumerated_number_set enables discontinuous sets of numbers, either all real numbers or integer numbers or a mixture of integer and real numbers to be defined (see 5.2.5.4).

EXAMPLE Figure 158 shows the **integer_numbers** 45, 59, 73 are members of an **enumerated_number_set**. No order is implied.

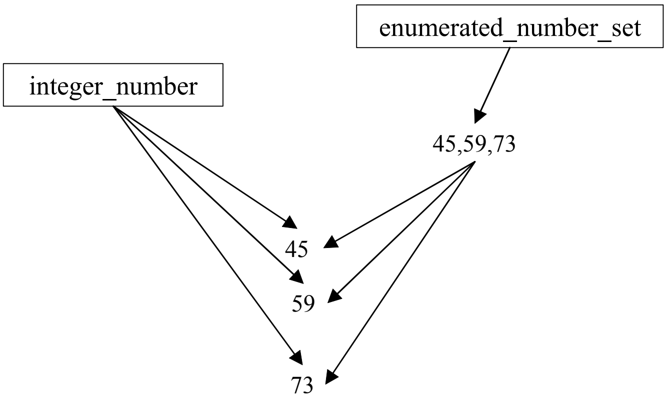


Figure 158 — Enumerated number set

4.8.5.2.2 Number range

number_ranges are constrained single-dimension **number_spaces** (see 5.2.5.9). The upper and lower bounds of the range are particular members of the **number_range**. The model for this is shown in Figure 159.

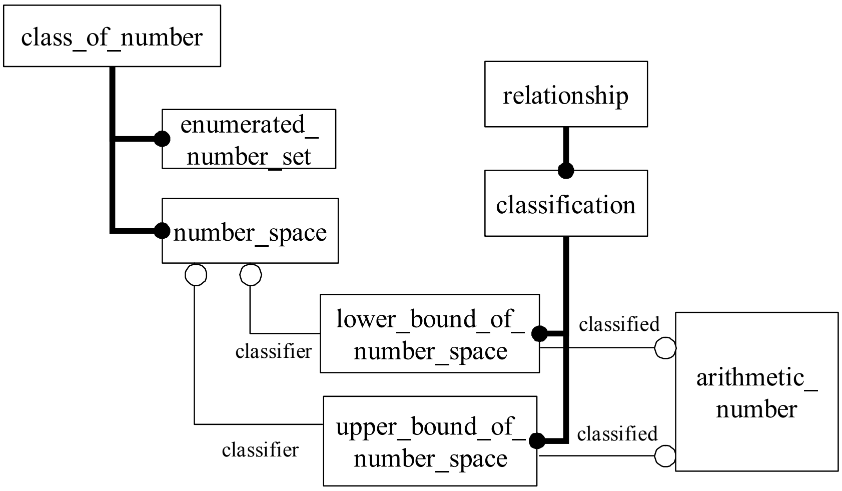


Figure 159 — Bounds of number range

EXAMPLE Figure 160 shows the **real_numbers** in the range 5.2 to 9.3 is a **number_range** with lower bound of 5.2 and an upper bound of 9.3.

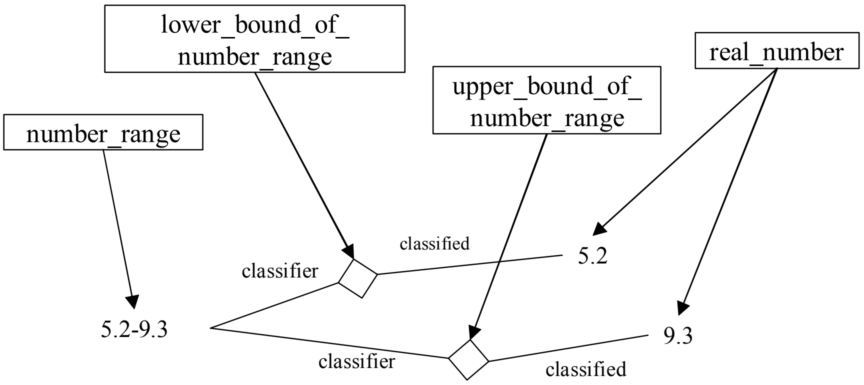


Figure 160 — Number range 5.2 to 9.3

4.8.5.2.3 Multidimensional number spaces

A **multidimensional_number_space** is a continuum of **multidimensional_number**s (see 5.2.5.8, Figure 180 and Figure 181).

EXAMPLE 1 Figure 161 shows 'R1', the continuum of all real numbers, is a **number_space**. 'R3', the three-dimensional real **number_space**, is a **multidimensional_number_space**, with 'R1' as elements 1, 2 and 3.

NOTE 'R1' and the entity type **real_number** of this model are the same thing.

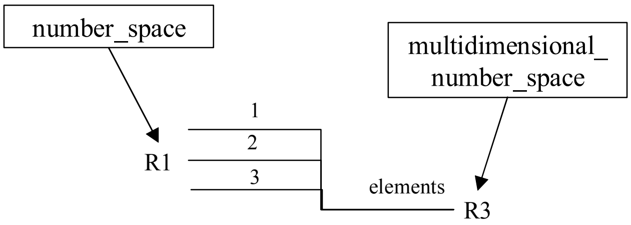


Figure 161 — R3 real number space

EXAMPLE 2 Figure 162 shows that 'Complex number' is a **multidimensional_number_space**. The complex number written as 7.1+ 9.3i is a **multidimensional_number** with a real part 7.1 and an imaginary part 9.3. The elements of 'Complex number' are defined by the **role_and_domain**s 'Real part' and 'Imaginary part'. These are combinations of the real or imaginary role with the **real_number** domain. The classification of [7.1;9.3] by 'Complex number' determines the roles of the two numbers.

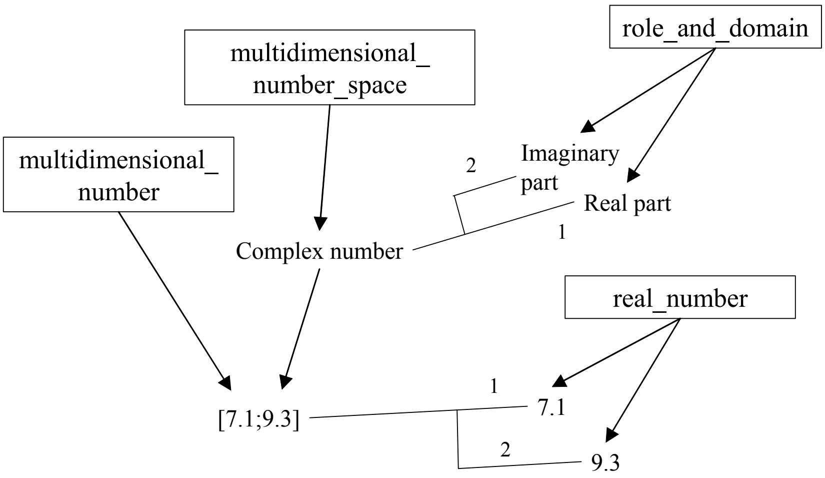


Figure 162 — Complex numbers

4.9 Functional mapping

functional_mapping is a type of **relationship** that maps an input to a result (see 5.2.15.3 and Figure 191). Such mappings can be one-to-one or many-to-one. The former are known as isomorphic mappings. Sets or classes of functional mapping relationships form functions.

The model elements for **functional_mapping** are shown in Figure 163.

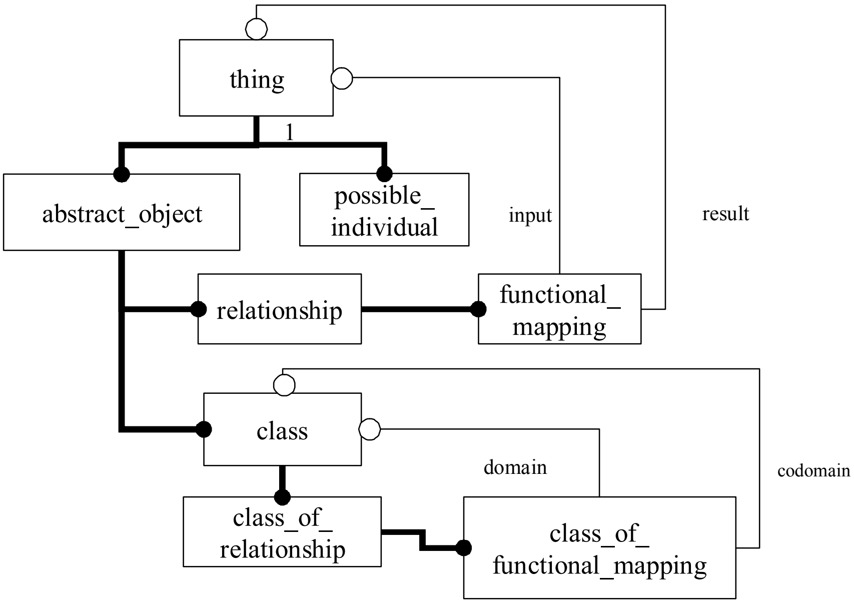


Figure 163 — Functional mapping

EXAMPLE 1 Figure 164 shows data for the function ‘x^2’ where ‘x’ is any **real_number**.

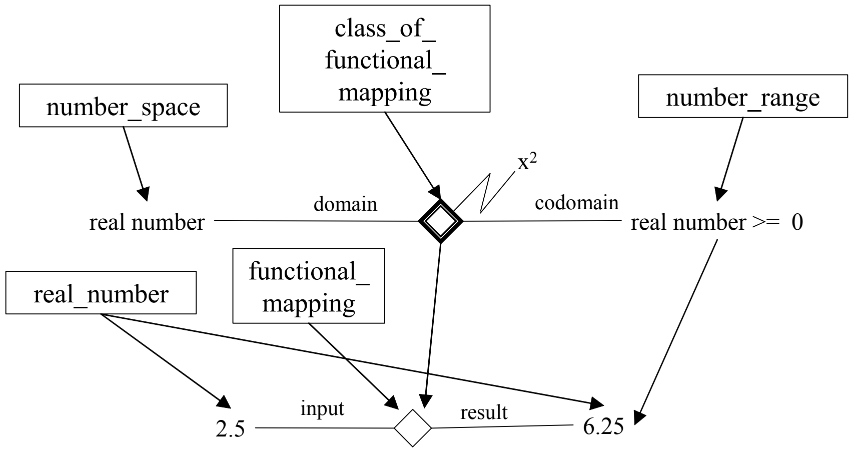


Figure 164 — X^2 functional mapping

functional_mappings are not restricted to numbers.

EXAMPLE 2 Figure 165 shows a pressure 'difference' **functional_mapping**. 'a' is the 'upstream' pressure and 'b' the 'down stream' pressure of flow across a filter. Pressure 'c' is the 'difference' of pressures 'a' and 'b', where 'a' is the 'reference' and 'b' is the 'differand'. The 'difference' **class_of_functional_mapping** links a **class_of_multidimensional_object** consisting of two **role_and_domain**s, the input arguments, with the difference output **role_and_domain**. The mapping is not isomorphic as pressure 'c' may be the result of many combinations 'a' and 'b'.

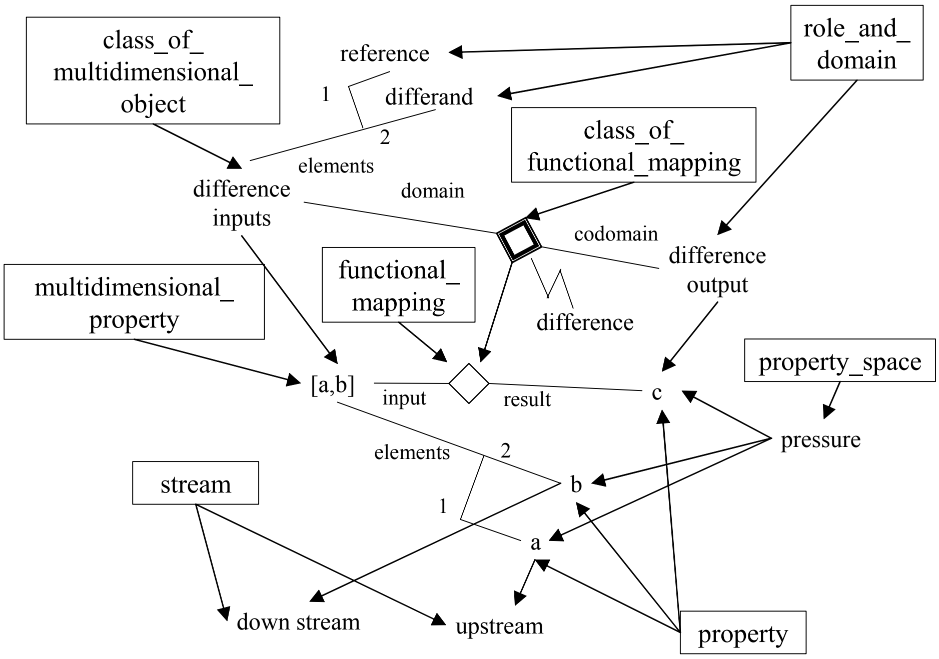


Figure 165 — Pressure difference functional mapping

Some classes of functional mapping are modelled as explicit subtypes of **functional_mapping** (see Figure 191). These are shown in Figure 166 below. Three of the subtypes deal with the set operations: intersection; union; and difference (see 5.2.25). **property_quantification** and its class counterpart **scale** is described in 4.8.4.3.2.

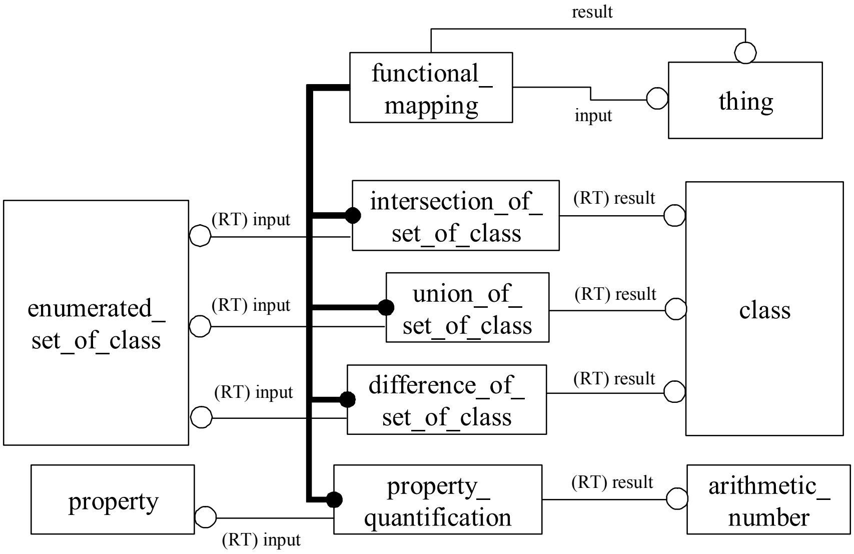


Figure 166 — Functional mapping subtypes

EXAMPLE 3 Figure 167 shows three Venn diagrams defining: class ‘I’ that is the intersection of classes ‘A’, ‘B’ and ‘C’.; class ‘U’ that is the union of classes ‘A’,’B’, and ‘C’; and class ‘D’ that is the difference of classes ‘A’, ‘B’ and ‘C’. The model representation by **functional mapping** types is shown in Figure 168.

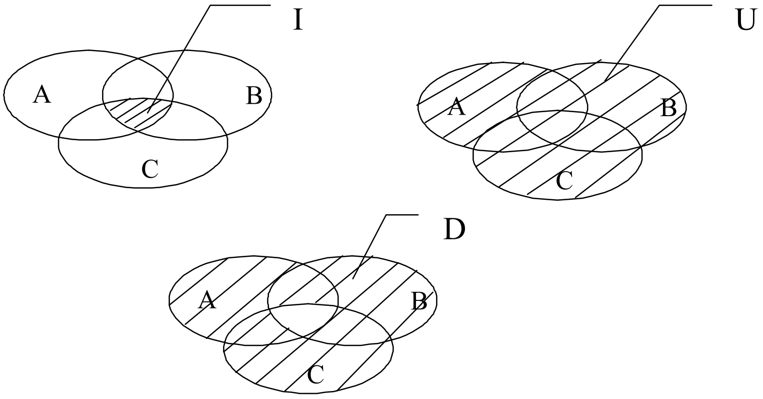


Figure 167 — Venn diagrams of classes A,B,C, I, U, and D

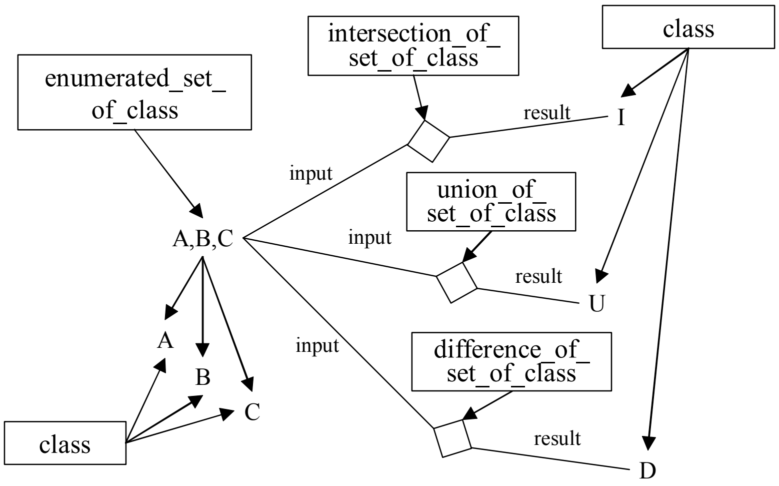


Figure 168 — Intersection, union and difference of classes A, B, C

4.10 Other user-defined relationships

4.10.1 Other relationship

Many types of relationship are not explicitly modelled. To allow such relationships to be represented, the entity type **other_relationship** has been defined (see 5.2.11.1 and Figure 187). The model for this is illustrated in Figure 169.

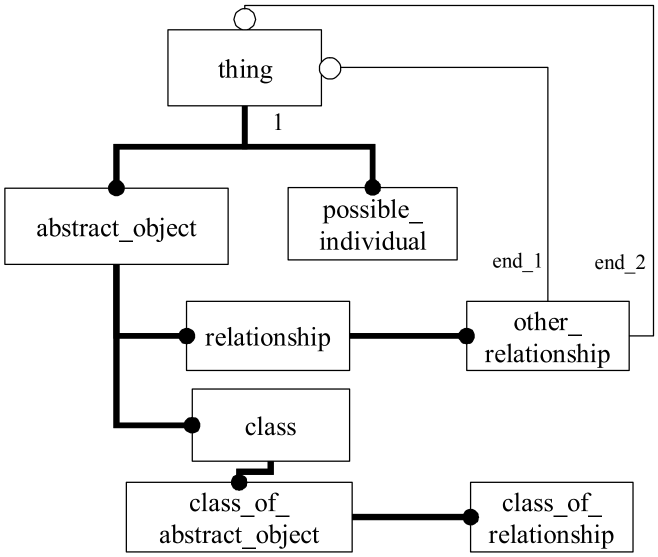


Figure 169 — Other relationship

Any two **thing**s can be involved in an **other_relationship**. They are distinguished by the roles **end_1** and **end_2**. **other_relationship** excludes relationships that are members of the other explicit subtypes of **relationship**. The significance or meaning of an **other_relationship** may be given by classifying it with one or more **class_of_relationship** (see 5.2.12 and Figure 188).

EXAMPLE Figure 170 shows an ordered pair of individuals consisting of myself for a period of time and a car for the same period of time that is an **other_relationship** classified as an ownership relationship. As yet there is no indication of whether I own the car or the car owns me.

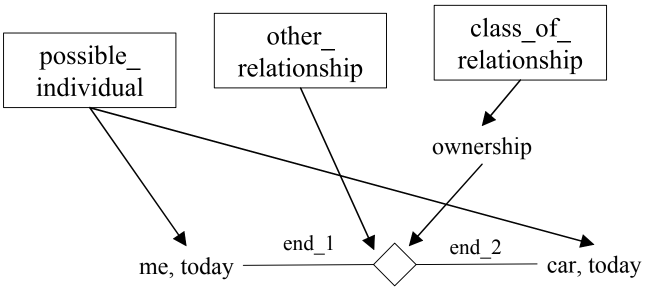


Figure 170 — Ownership relationship

4.10.2 Class of relationship with signature

Types of **other_relationship** may be classified using **class_of_relationship_with_signature** (see 5.2.13.2 and Figure 189). These enable the significance or meaning of the member relationships, the roles of the member relationship participants, and constraints on the domain or types of the member relationship participants, to be specified.

The model for **class_of_relationship_with_signature** is shown in Figure 171. **role_and_domain** classes are used to constrain the two ends of the member relationships.

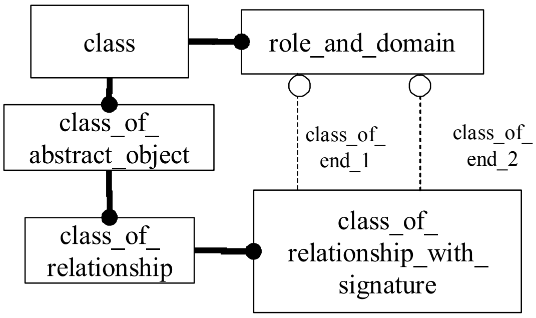


Figure 171 — Class of relationship with signature

EXAMPLE 1 The ‘owner’ and ‘owned’ roles omitted from EXAMPLE 1 may be defined by classifying the **other_relationship** as an ownership **class_of_relationship_with_signature**, with the roles of ‘owner’ and ‘owned’. Figure 172 shows the data for this. ‘Owner’ and ‘Owned’ are both members of **role**, as they are not constrained by domain. Also, both the **possible_individual**s are members of the respective roles.

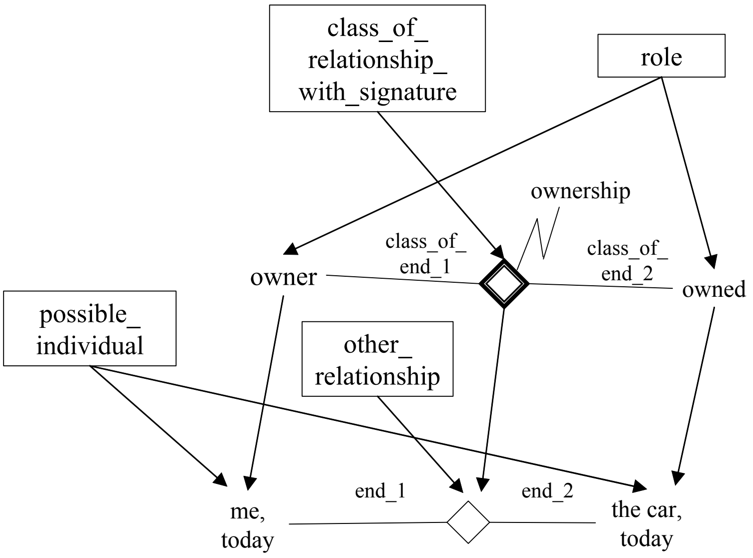


Figure 172 — Ownership class of relationship with signature

EXAMPLE 2 Figure 173 shows the class of relationship "insertion of individual" modelled using **class_of_relationship_with_signature**. **class_of_end_1** refers to the **role_and_domain** "inserted & individual" and **class_of_end_2** refers to the **role_and_domain** "host & individual". The **other_relationship**, linking the **physical_object**s #1234 and #AC6756 is a member of the 'insertion of individual' **class_of_relationship**.

The **class_of_relationship** 'pipe thermowell insertion' is shown as a specialization of 'insertion of individual'. The specialization restricts the domain of **class_of_end_2** to be 'pipeline's and the domain of **class_of_end_1** to 'thermowell's, instead of any individual.

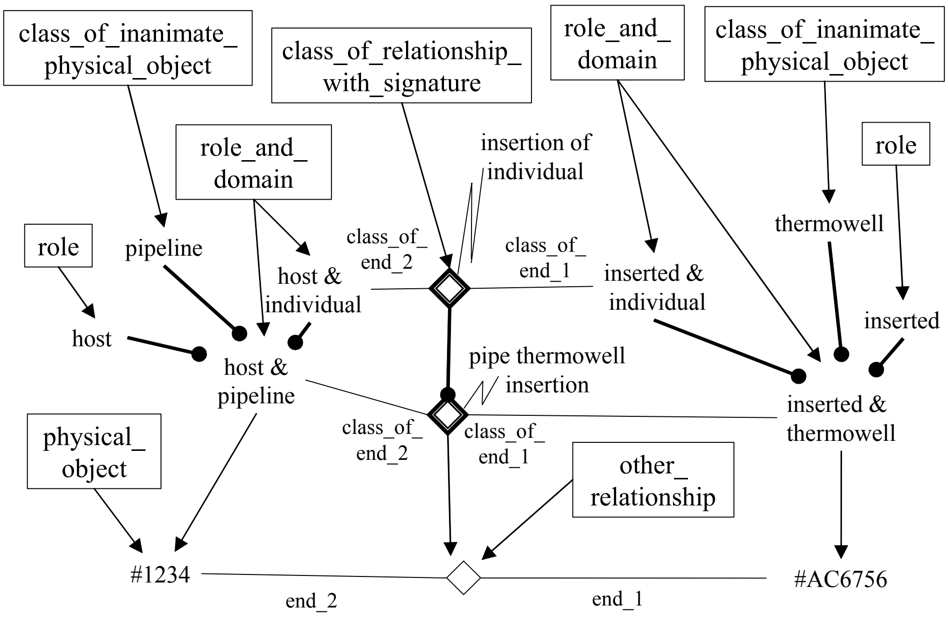


Figure 173 — Insertion of individual

The **physical_object** #1234 is a member of the **role_and_domain** ‘host & pipeline’, which is a combination of the **role** of ‘host’ and the domain of ‘pipeline’. ‘Host & pipeline’ is a subset (specialization) of the **role_and_domain** ‘host & individual’. The **physical_object** #AC6756 is a member of the **role_and_domain** ‘inserted & thermowell’, which is a specialization the **role** ‘inserted’ and the domain ‘thermowell’.

4.10.3 Cardinality constraints

Cardinality constraints may also be applied to the members of **class_of_relationship_with_signature**.

EXAMPLE Figure 174 shows the use of the cardinality constraints with **class_of_relationship_with_signature** to define the **class_of_inanimate_physical_object** ‘6 of M8 bolts’, each member being 6 bolts. The **class_of_relationship_with_signature** ‘6 M8 bolt assembly’ has a cardinality such that each ‘6 of M8 bolts’ is linked by exactly 6 relationships to different M8 bolts at all times. The cardinality of a pack for bolts is zero to one, as a bolt may not be in a pack of six, but can only be in one such pack at a time. The **class_of_relationship_with_signature** ‘6 M8 bolt assembly’ is a specialization of the more general **class_of_relationship** ‘assembly of individual’.Figure 174 also shows one of these relationships for a particular 6 pack and a particular bolt.

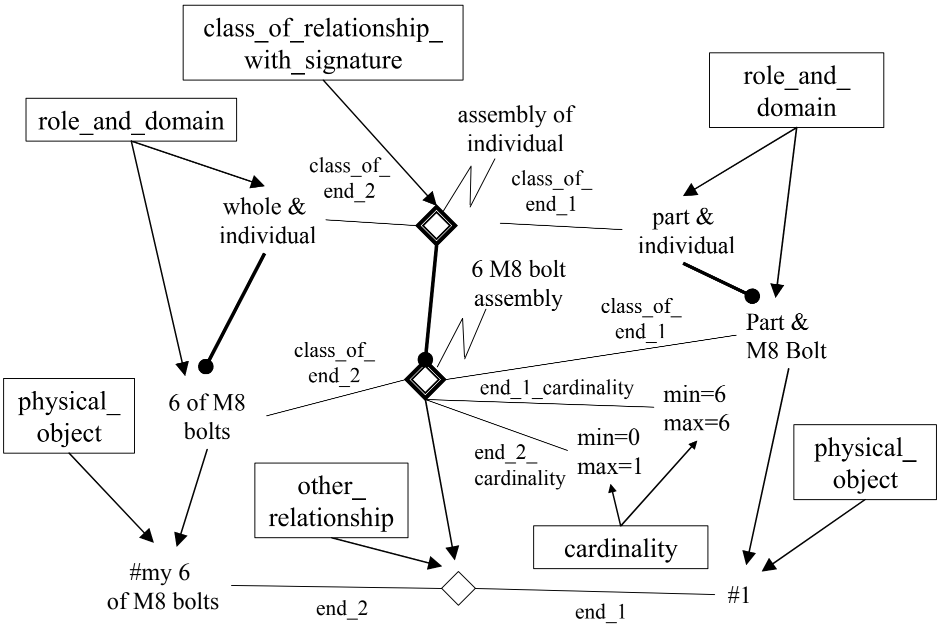


Figure 174 — 6 of M8 bolts

4.10.4 Assymetric other relationship classes

class_of_relationship_with_signature may be combined with **class_of_relationship_with_related_end_1** or **class_of_relationship_with_related_end_2** to recognize asymmetric non-explicit **class_of_relationship**s (see 5.2.12.3, 5.2.12.4, 5.2.13.2, and Figure 188). The model elements for this are shown in Figure 175.

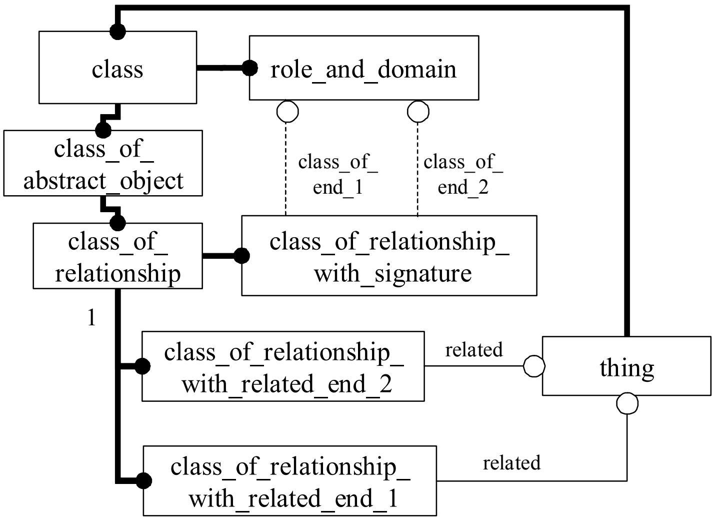


Figure 175 — Asymmetric class of relationship with signature

EXAMPLE Figure 176 shows the use of **class_of_relationship_with_signature** to define the 'fabrication' **class_of_relationship**. 'fabrication' relationships link a 'manufacturer' to a 'product'. A second **class_of_relationship_with_signature** that is also a **class_of_relationship_with_related_end_1** is used to define the specialization of the 'fabrication' relationships where the 'fabrication' is performed by 'Bloggs & Co'. The **other_relationship**, linking the individuals #1234 and 'Bloggs & Co', is a member of the **class_of_relationship** 'fabrication by Bloggs'.

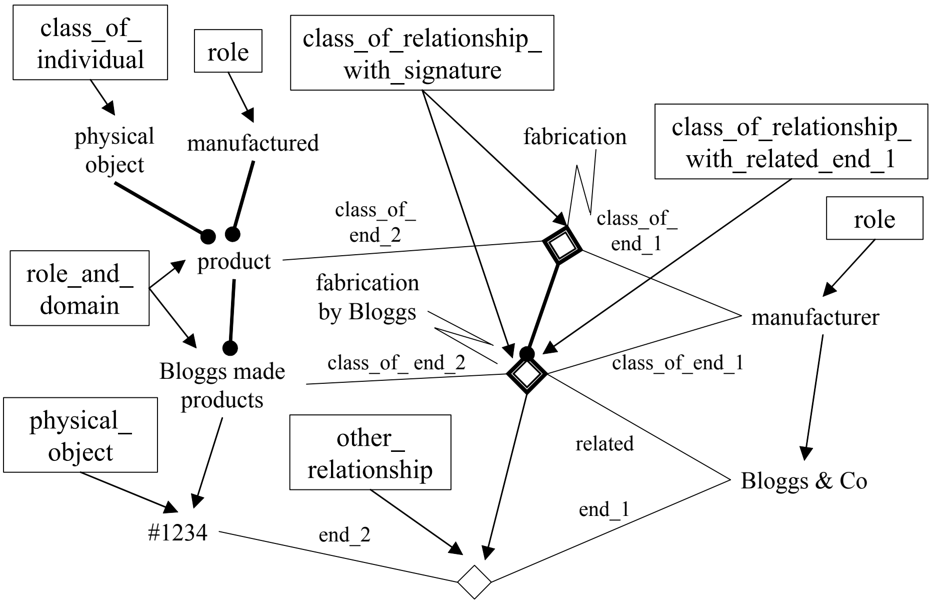


Figure 176 — Bloggs made products

5 Lifecycle integration schema

5.1 Introduction

This clause specifies the schema supporting life-cycle integration. This clause is divided into a number of subclauses. This division into subclauses is presentational in nature only. The subject areas described in each subclause are not separate or separable schemas.

NOTE 1 A listing of the complete EXPRESS schema specified in this part of ISO 15926, without comments or other explanatory text, is available from the Internet - see Annex B.

NOTE 2 This schema does not use all the facilities of the EXPRESS language. Annex C lists facilities that are not used.

5.2 Schema definition

The following EXPRESS declaration begins the **lifecycle_integration_schema**.

<u>EXPRESS specification:</u>

```
*)
SCHEMA lifecycle_integration_schema;
(*
```

5.2.1 Things

This subclause contains the declaration of the entity data type **thing**, which is the root entity data type of the **lifecycle_integration_schema**.

NOTE Figure 177 is a diagram of the entity data type(s) defined in this subclause (see also 4.6.1).

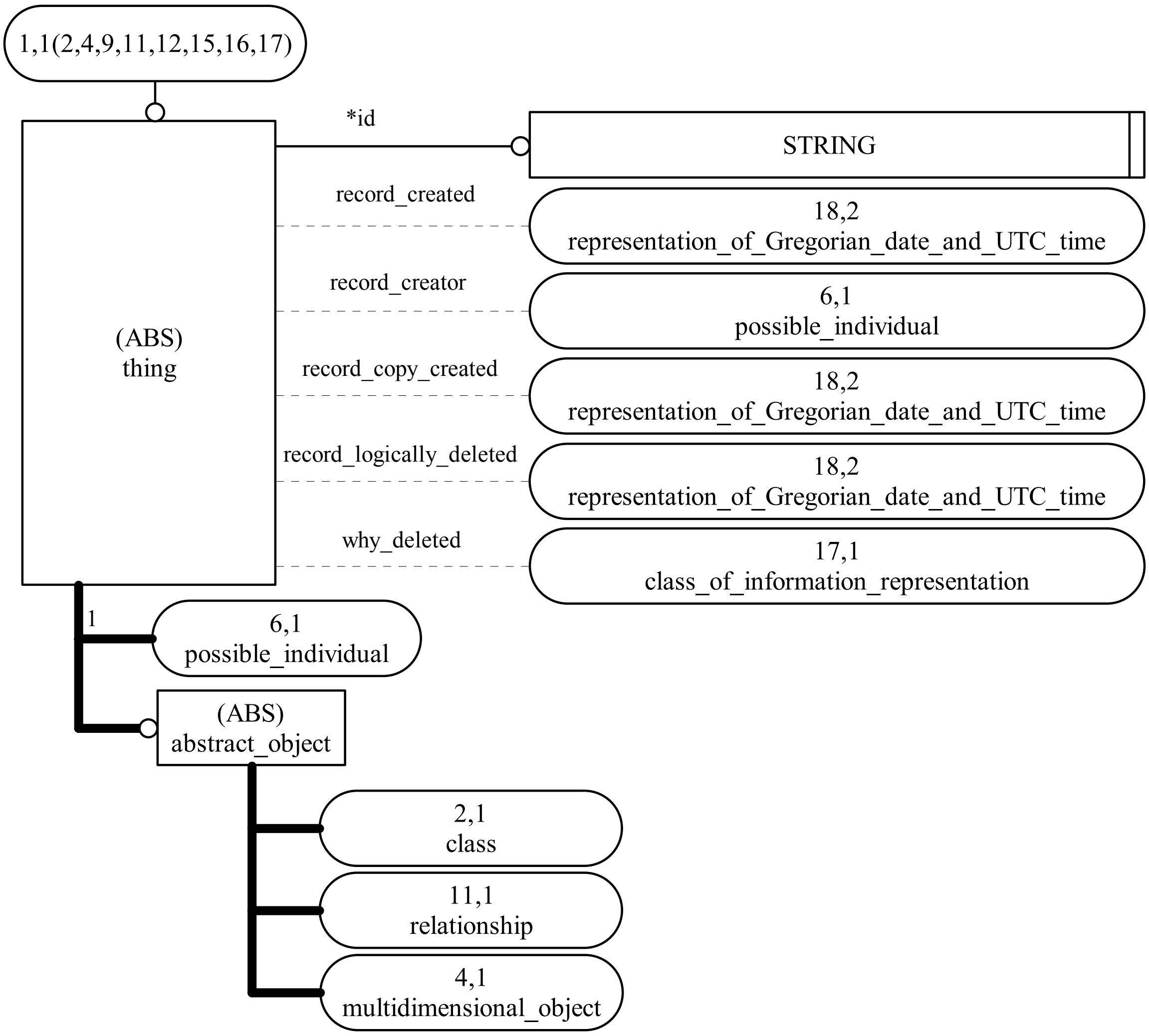


Figure 177 — lifecycle_integration_schema EXPRESS-G diagram 1 of 29

5.2.1.1 abstract_object

An **abstract_object** is a **thing** that does not exist in space-time.

EXPRESS specification:

```
*)
  ENTITY abstract_object
    ABSTRACT SUPERTYPE
    SUBTYPE OF(thing);
  END_ENTITY;
(*
```

5.2.1.2 thing

A **thing** is anything that is or may be thought about or perceived, including material and non-material objects, ideas, and actions.

Every **thing** is either a **possible_individual**, or an **abstract_object**.

NOTE 1 Every **thing** is identifiable within a system. System identifiers created by other systems and received as part of a data exchange may be stored for future reference as an **identification**, referring to the originating organisation or system.

NOTE 2 Every example provided for other entity data types declared in this schema is also an example of **thing**.

EXPRESS specification:

```
*)
  ENTITY thing
    ABSTRACT SUPERTYPE OF (ONEOF(possible_individual,
                    abstract_object));
    id                  : STRING;
    record_copy_created : OPTIONAL
                          representation_of_Gregorian_date_and_UTC_time;
    record_created      : OPTIONAL
                          representation_of_Gregorian_date_and_UTC_time;
    record_creator      : OPTIONAL possible_individual;
    record_logically_deleted
                        : OPTIONAL
                          representation_of_Gregorian_date_and_UTC_time;
    why_deleted         : OPTIONAL class_of_information_representation;
  UNIQUE
    UR1      :id;
  END_ENTITY;
(*
```

Attribute definitions:

id : an identifier of the **thing** for the purposes of record management within a system

record_copy_created : the date and time when this copy of the record was created in the current system. This attribute shall have a value only when the current system is not the originating system.

record_created : the date and time on which this record was first created in its originating system

record_creator : the person, organization or system that first created this record in the originating system

record_logically_deleted : the date and time that this record was logically deleted

why_deleted : the reason why the record was logically deleted

NOTE Logical deletion means that whilst the record is still available in the system as a matter of historical record, it is no longer considered a valid statement. That is to say it is considered that it was never true.

Formal proposition

UR1 : the id of the **thing** shall be unique within a system

5.2.2 Classes

This subclause contains the declarations of entity data types that represent classes.

NOTE Figure 178 is a diagram of the entity data type(s) defined in this subclause (see also 4.6.3 and 4.8).

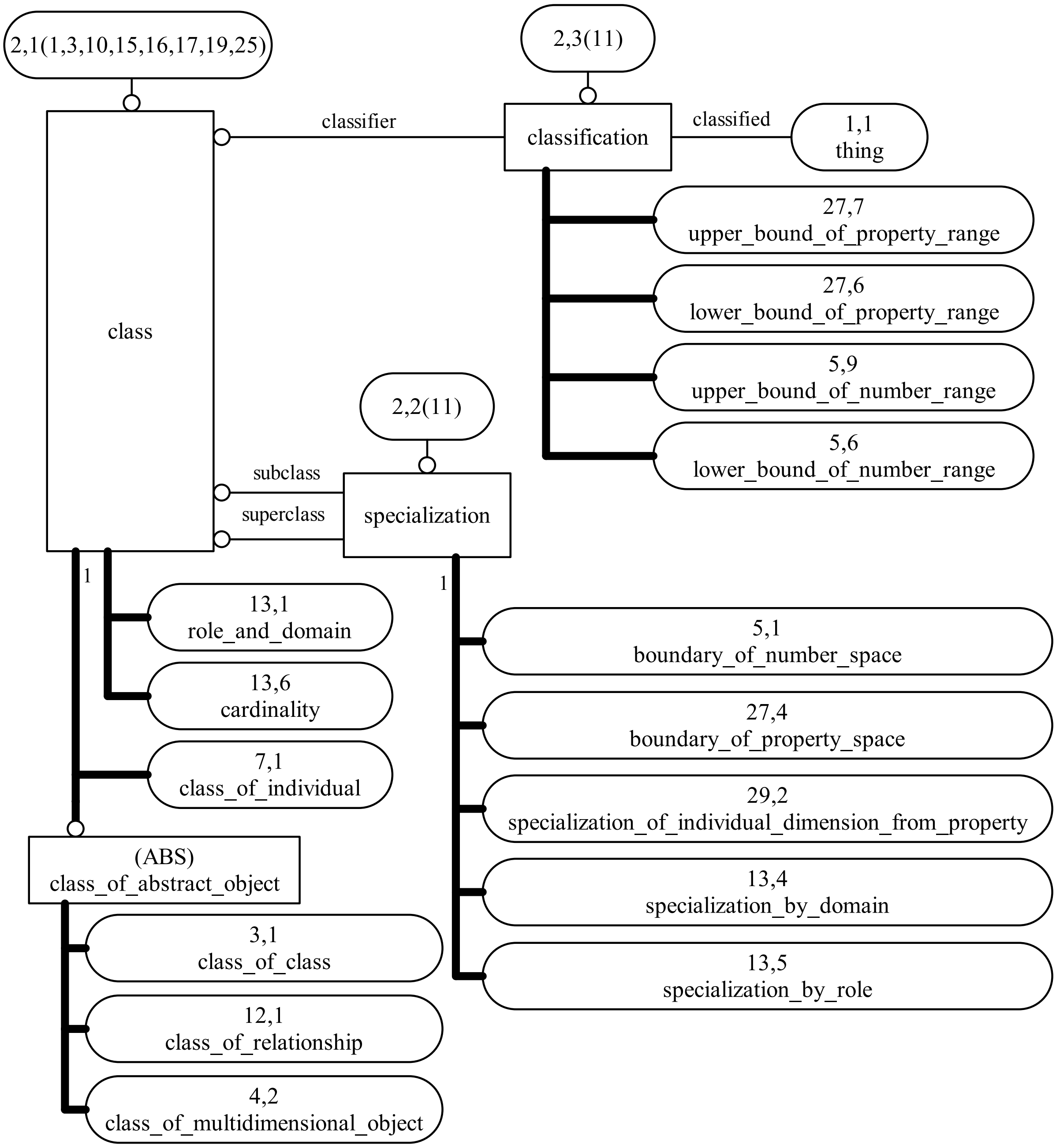


Figure 178 — lifecycle_integration_schema EXPRESS-G diagram 2 of 29

5.2.2.1 class

A **class** is a **thing** that is an understanding of the nature of things and that divides things into those which are members of the class and those which are not according to one or more criteria.
The identity of a **class** is ultimately defined by its members. No two classes have the same membership. However, a distinction must be made between a **class** having members, and those members being known, so within an information system the members recorded may change over time, even though the true membership does not change.

NOTE 1 The membership of a **class** is unchanging as a result of the spatio-temporal paradigm upon which this schema is based. In another paradigm it might be stated that a car is red at one time, and green at another time, indicating that the class of red things and class of green things changed members. However, using a spatio-temporal paradigm, a temporal part, state 1, of the car was red, and another temporal part of the car, state 2, was green. In this way the members of the classes red and green are unchanging. The same principle applies to future temporal parts as to past temporal parts, it is just more likely that the membership of these is not known.

A **class** may be a member of another **class** or of itself.

NOTE 2 The set theory that applies to classes in this model is non-well-founded set theory [3] (see D.2.4). This permits statements like "class is a member of class", unlike traditional set theories such as Zermelo-Fraenkel set theory found in standard texts [4].

There is a null **class** that has no members.

NOTE 3 The known members of a **class** are identified by **classification**.

EXAMPLE 1 ‘Centrifugal pump’ is a **class**.

EXAMPLE 2 ‘Mechanical equipment type’ is a **class**.

EXAMPLE 3 ‘Temperature’ is a **class**.

EXAMPLE 4 ‘Commercial fusion reactor’ is a **class**.

NOTE 4 Although there is only one **class** that has no members, there can be a **class** that has no members in the actual world, but which does have members in other possible worlds.

EXAMPLE 5 ‘Centigrade scale’ is a **class**.

EXPRESS specification:

```
*)
  ENTITY class
    SUPERTYPE OF (role_and_domain ANDOR cardinality ANDOR
                    ONEOF(class_of_individual,
                    class_of_abstract_object))
    SUBTYPE OF(abstract_object);
  END_ENTITY;
 (*
```

5.2.2.2 class_of_abstract_object

A **class_of_abstract_object** is a **class** whose members classify members of **abstract_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_abstract_object
    ABSTRACT SUPERTYPE
    SUBTYPE OF(class);
  END_ENTITY;
(*
```

5.2.2.3 classification

A **classification** is type of **relationship** that indicates that the classified **thing** is a member of the classifier **class**.

Classification is not transitive.

NOTE A subtype of **relationship** is transitive if when A is related to B, and B is related to C in the same way, then A is necessarily related to C in that way. **specialization** and **composition** are examples of transitive subtypes of **relationship**. However, because **classification** is not transitive does not mean that A cannot be related to C in the same way, only that it does not necessarily follow from A being related to B and B being related to C.

EXAMPLE 1 The **relationship** that indicates that 'London' is a member of the class known as 'capital city' is a **classification**.

EXAMPLE 2 The **relationship** that indicates that 'pump' is a member of the class 'equipment type' is a **classification**.

EXPRESS specification:

```
*)
  ENTITY classification
    SUBTYPE OF(relationship);
    classified                  :thing;
    classifier                  :class;
  END_ENTITY;
(*
```

Attribute definitions:

classified : the **thing** that is a member of the classifier **class**

classifier : the **class** of which the classified **thing** is a member

5.2.2.4 specialization

A **specialization** is a **relationship** that indicates that all members of the subclass are members of the superclass. **specialization** is transitive.

NOTE If A is a **specialization** of B and B is a **specialization** of C, then A is necessarily a **specialization** of C.

EXAMPLE 'Centrifugal pump' is a **specialization** of 'pump'.

EXPRESS specification:

```
*)
  ENTITY specialization
 SUPERTYPE OF (ONEOF(
                  boundary_of_number_space,
                  boundary_of_property_space,
                  specialization_by_domain,
                  specialization_by_role,
                  specialization_of_individual_dimension_from_property)
                  )
    SUBTYPE OF(relationship);
    subclass                    :class;
    superclass                  :class;
  END_ENTITY;
 (*
```

Attribute definitions:

subclass : the **class** that is a specialization of the superclass **class**

superclass : the **class** that is a generalization of the subclass **class**

5.2.3 Classes of class

This subclause contains the declarations of entity data types that represent classes of class.

NOTE Figure 179 is a diagram of the entity data type(s) defined in this subclause.

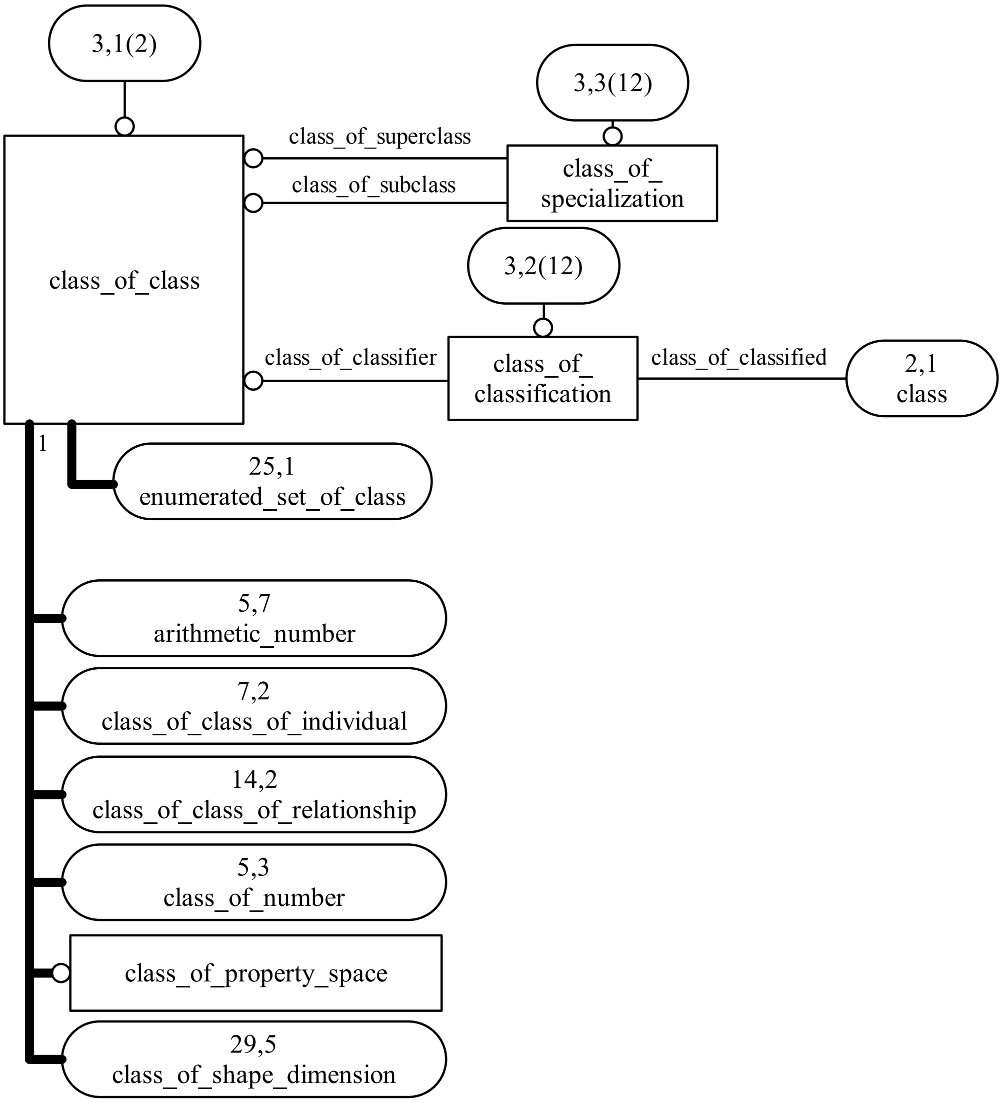


Figure 179 — lifecycle_integration_schema EXPRESS-G diagram 3 of 29

5.2.3.1 class_of_class

A **class_of_class** is a **class** whose members are instances of **class**.

NOTE When it is necessary to classify a **class_of_class**, another **class_of_class** can be used. This is because a **class_of_class** is a **class**.

EXPRESS specification:

```
*)
  ENTITY class_of_class
    SUPERTYPE OF (ONEOF(
                        arithmetic_number,
                        class_of_class_of_individual,
                        class_of_class_of_relationship,
                        class_of_number,
                        class_of_property_space,
                        class_of_shape_dimension)
                        ANDOR enumerated_set_of_class)
    SUBTYPE OF(class_of_abstract_object);
  END_ENTITY;
(*
```

5.2.3.2 class_of_classification

A **class_of_classification** is a **class_of_relationship** whose members are members of **classification**. A **class_of_classification** indicates that a member of the **class_of_classified class** is classified by one or more members of the **class_of_classifier class_of_class**.

EXAMPLE The link between **class** 'centrifugal pump' and the **class_of_property** 'RPM', indicating that a 'centrifugal pump' is a member of at least one 'RPM' **class**, can be represented by an instance of **class_of_classification**.

EXPRESS specification:

```
*)
  ENTITY class_of_classification
    SUBTYPE OF(class_of_relationship);
    class_of_classified        :class;
    class_of_classifier        :class_of_class;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_classified : the **class** that is the class_of_classified in the **class_of_classification**

class_of_classifier : the **class_of_class** that is the class_of_classifier in the **class_of_classification**

5.2.3.3 class_of_property_space

A **class_of_property_space** is a **class_of_class** whose members are members of **property_space**.

EXAMPLE 1 Property curves, property areas, and property volumes of various dimensionality and degrees of freedom are members of **class_of_property_space**.

EXAMPLE 2 'Pump performance curve' is an example of **class_of_property_space**.

EXPRESS specification:

```
*)
  ENTITY class_of_property_space
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.3.4 class_of_specialization

A **class_of_specialization** is a **class_of_relationship** whose members are instances of **specialization**. It indicates that a member of the **class_of_subclass** is a subclass of a member of the **class_of_superclass**.

EXAMPLE The **class_of_specialization** that indicates that members of the class "family of ASME bolts", e.g. 3 inch, 2 inch bolts, are specializations of members of the **enumerated_property_set** "set of bolt lengths", e.g. 3 inch, 2 inch.

EXPRESS specification:

```
*)
  ENTITY class_of_specialization
    SUBTYPE OF(class_of_relationship);
    class_of_subclass          :class_of_class;
    class_of_superclass        :class_of_class;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_subclass : the **class_of_class** whose members are the subclass in the members of the **class_of_specialization**

class_of_superclass : the **class_of_class** whose members are the superclass in the members of the **class_of_specialization**

5.2.4 Multidimensional objects

This subclause contains the declarations of entity data types that represent multidimensional objects.

NOTE Figure 180 is a diagram of the entity data type(s) defined in this subclause (see also 4.6.5, 4.8.4.3.6, and 4.8.5.2.3).

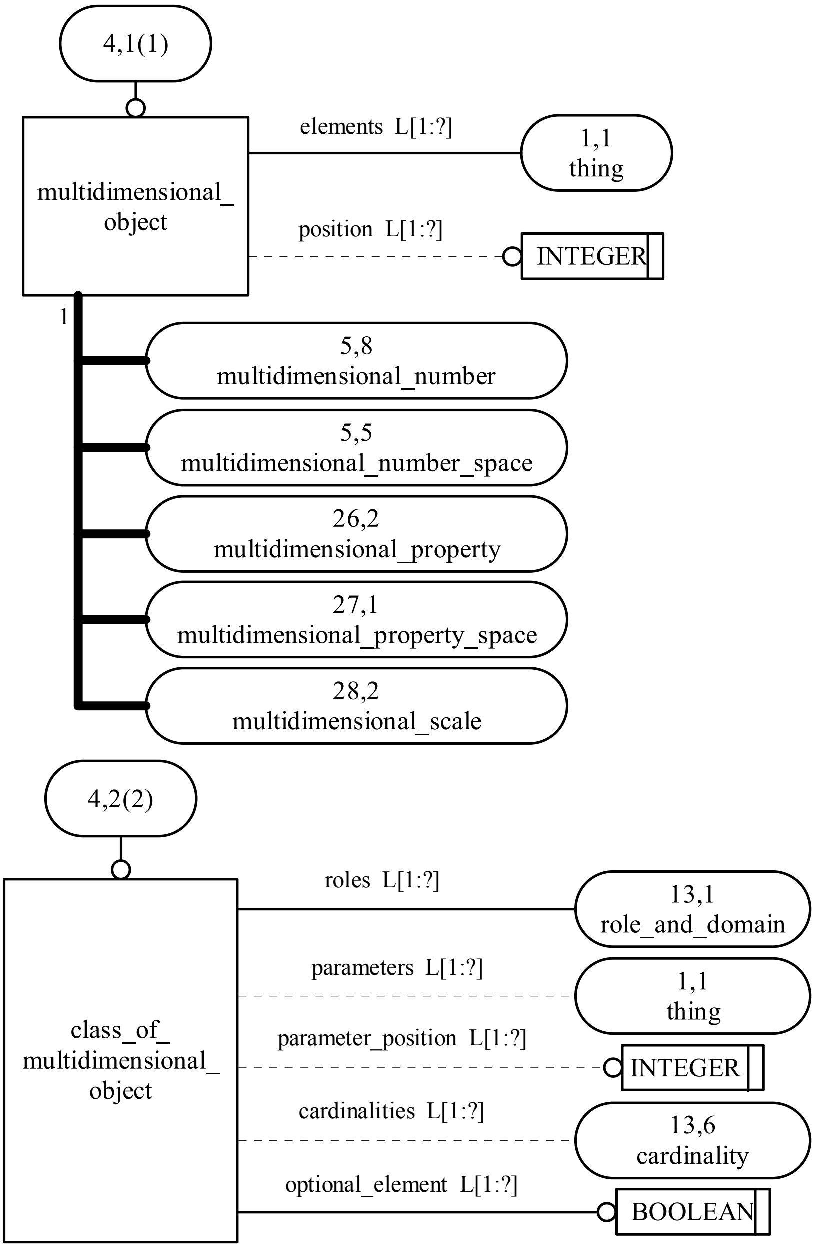


Figure 180 — lifecycle_integration_schema EXPRESS-G diagram 4 of 29

5.2.4.1 class_of_multidimensional_object

A **class_of_multidimensional_object** is a **class** whose members are instances of **multidimensional_object**. The role played by each position in the classified **multidimensional_object** is specified at the same position in the **roles** attribute. Constant values that apply to any position in **roles** are specified in the same position in the **parameters** attribute. The cardinalities for the **roles** attribute are specified by the same position in the **cardinalities** attribute.

EXAMPLE The definition of the input to a function y = a + bx to convert Celsius to Fahrenheit with roles [a, b, x] defining the input **multidimensional_object**, and **parameters** list [32,1.8] with **parameter_position** list [1,2] is an example of a **class_of_multidimensional_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_multidimensional_object
    SUBTYPE OF(class_of_abstract_object);
    cardinalities               : OPTIONAL LIST [1:?] OF cardinality;
    optional_element            : LIST [1:?] OF BOOLEAN;
    parameters                  : OPTIONAL LIST [1:?] OF thing;
    parameter_position          : OPTIONAL LIST [1:?] OF INTEGER;
    roles                       : LIST [1:?] OF role_and_domain;
  END_ENTITY;
(*
```

Attribute definitions:

cardinalities : the list of cardinalities that apply to the roles. If no cardinality is specified, then there are no constraints on the cardinality. If the cardinality is specified, then it shall be specified for all roles.

optional_element : indicates if the element in this list position in a **multidimensional_object** that is a member of this **class_of_multidimensional_object** is optional (or mandatory). The value TRUE means it is optional, the value FALSE means it is mandatory.

parameters : the list of parameters associated with the roles

parameter_position : the list of positions relative to the roles for the list of parameters

NOTE 2 This attribute is necessary because the EXPRESS LIST datatype does not allow empty positions. The list of positions provides the mapping to the role positions.

roles : the roles associated with the classified **multidimensional_object**

5.2.4.2 multidimensional_object

A **multidimensional_object** is an **abstract_object** that is an ordered list of **thing**. The significance of the **multidimensional_object** is determined by being a member of a **class_of_multidimensional_object** that indicates the role played by each of its elements.

NOTE The **multidimensional_object** [A,B,C] is different from [B,C,A].

EXAMPLE [32, 1.8, 20] is a **multidimensional_object** that may be specified to be the input parameters for the function y=a+bx to convert 20 Celsius to Fahrenheit.

EXPRESS specification:

```
*)
  ENTITY multidimensional_object
    SUPERTYPE OF (ONEOF(multidimensional_property_space,
                        multidimensional_number,
                        multidimensional_property,
                        multidimensional_number_space,
                        multidimensional_scale))
    SUBTYPE OF(abstract_object);
    elements                    : LIST [1:?] OF thing;
    position                    : OPTIONAL LIST [1:?] OF INTEGER;
  END_ENTITY;
(*
```

Attribute definitions:

elements : the list of **thing** that constitute the **multidimensional_object**. The role of each **thing** is determined by a classifying **class_of_multidimensional_object**.

position : the position of the element relative to the list of roles in the classifying **class_of_multidimensional_object**. The elements shall be listed in ascending order. This attribute is required when some elements are missing. The EXPRESS list data type does not allow empty elements in the list. This attribute, when present, supplies the mapping information. When this attribute has no value, then all elements are present.

5.2.5 Numbers

This subclause contains the declarations of entity data types that represent numbers.

NOTE Figure 181 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.5).

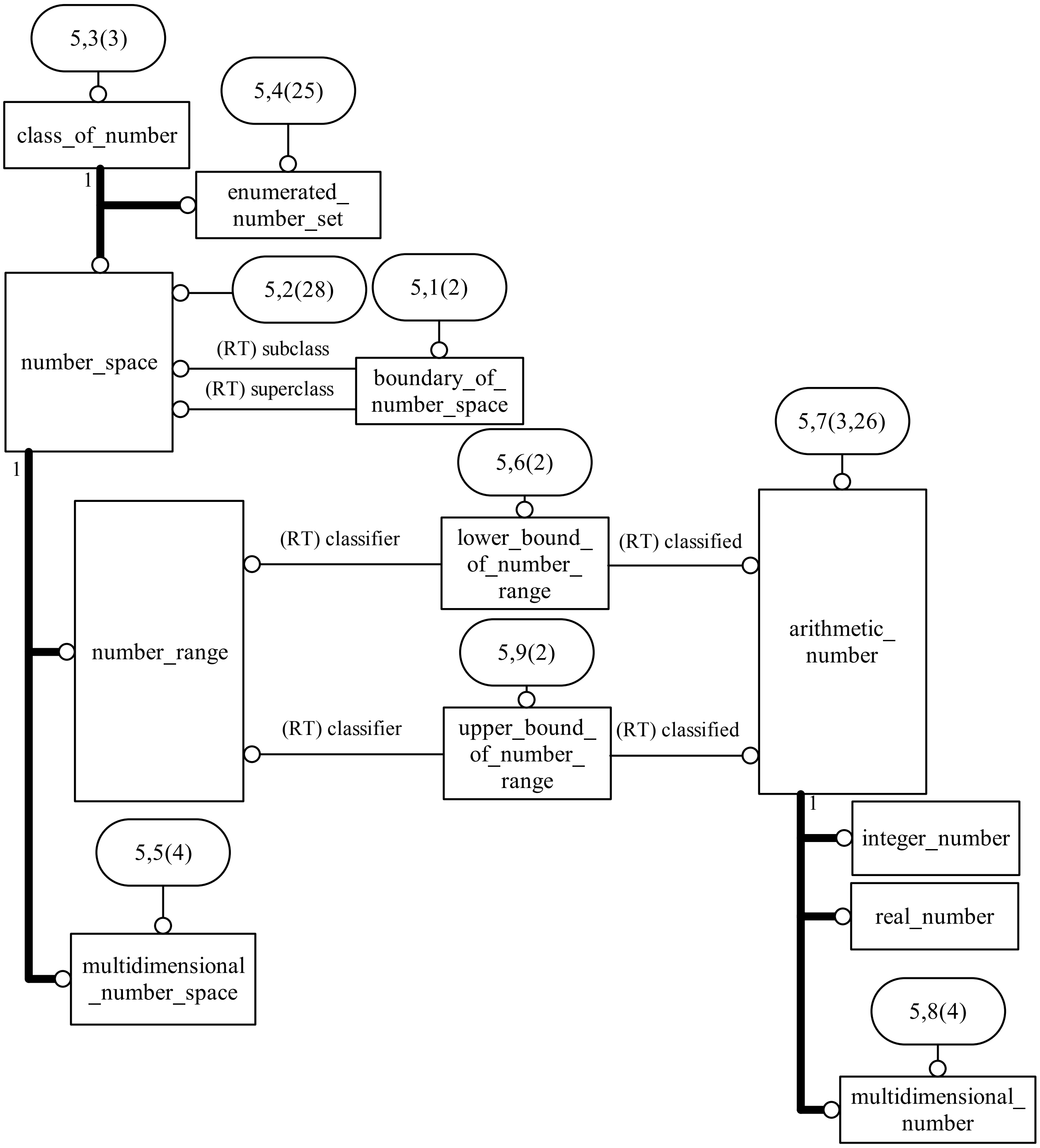


Figure 181 — lifecycle_integration_schema EXPRESS-G diagram 5 of 29

5.2.5.1 arithmetic_number

An **arithmetic_number** is a **class_of_class** whose member classes have the same sign and count or magnitude. An **arithmetic_number** is the number itself, not any representation of the number.

NOTE **integer_number** is not a subtype of **real_number**. The members of **integer_number** are part of a different continuum from the members of **real_number**, but are isomorphic to a subset of it.

EXAMPLE 1 The number 2 and the number 2.0 can be represented by instances of **arithmetic_number**.

EXAMPLE 2 Fifteen, the number itself not the English word "fifteen", is an **arithmetic_number**. It could be represented by an **EXPRESS_integer** or but could also be represented by "XV", or a binary, or a hexadecimal representation.

EXPRESS specification:

```
*)
  ENTITY arithmetic_number
    SUPERTYPE OF (ONEOF(real_number, integer_number,
                        multidimensional_number))
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.5.2 boundary_of_number_space

A **boundary_of_number_space** is a **specialization** that indicates that a **number_space** is a boundary to another **number_space**.

EXAMPLE The side of a cube in R3 is a **number_space** that is a plane in R3 and that is a boundary to the **number_space** that is a cube in R3.

EXPRESS specification:

```
*)
  ENTITY boundary_of_number_space
    SUBTYPE OF(specialization);
    SELF\specialization.subclass: number_space;
    SELF\specialization.superclass      :    number_space;
  END_ENTITY;
(*
```

5.2.5.3 class_of_number

A **class_of_number** is a **class_of_class** whose members are members of **arithmetic_number**.

EXAMPLE The class of prime numbers can be represented by an instance of **class_of_number**.

EXPRESS specification:

```
*)
  ENTITY class_of_number
    SUPERTYPE OF (ONEOF(number_space, enumerated_number_set))
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.5.4 enumerated_number_set

An **enumerated_number_set** is a **class_of_number** and an **enumerated_set_of_class**.

EXAMPLE The set of integer numbers {3,4,5} can be represented by an instance of **enumerated_number_set**.

EXPRESS specification:

```
*)
  ENTITY enumerated_number_set
    SUBTYPE OF(class_of_number, enumerated_set_of_class);
  END_ENTITY;
(*
```

5.2.5.5 integer_number

An **integer_number** is an **arithmetic_number** that is an integer number.

EXAMPLE 1, 2, and 10 are representations of integer_numbers.

EXPRESS specification:

```
*)
  ENTITY integer_number
    SUBTYPE OF(arithmetic_number);
  END_ENTITY;
(*
```

5.2.5.6 lower_bound_of_number_range

A **lower_bound_of_number_range** is a relationship that indicates an **arithmetic_number** is the lowest value in a **number_range**.

EXAMPLE 3.1 is the lower bound of the range [3.1 to 5.3].

EXPRESS specification:

```
*)
  ENTITY lower_bound_of_number_range
    SUBTYPE OF(classification);
    SELF\classification.classified      :     arithmetic_number;
    SELF\classification.classifier      :     number_range;
  END_ENTITY;
(*
```

Attribute definitions:

classified : the **arithmetic_number** that as classified is the lower bound to the **number_range**

classifier : the **number_range** that as classifier is bounded by the **arithmetic_number**

5.2.5.7 multidimensional_number

A **multidimensional_number** is an **arithmetic_number** that is also a **multidimensional_object**.

EXAMPLE [3.2, 5.4, 55.6] is a **multidimensional_number**.

EXPRESS specification:

```
*)
  ENTITY multidimensional_number
    SUBTYPE OF(arithmetic_number, multidimensional_object);
  END_ENTITY;
(*
```

5.2.5.8 multidimensional_number_space

A **multidimensional_number_space** is a **number_space** and a **multidimensional_object**.

EXAMPLE R3, the space defined as being all the triples of real numbers (e.g. **1.0, 2.1, 5.4**), is a **multidimensional_number_space**.

EXPRESS specification:

```
*)
  ENTITY multidimensional_number_space
    SUBTYPE OF(number_space, multidimensional_object);
  END_ENTITY;
(*
```

5.2.5.9 number_range

A **number_range** is a one-dimensional **number_space**.

EXAMPLE The **number_space** -273.1 to +infinity is a **number_range**.

EXPRESS specification:

```
*)
  ENTITY number_range
    SUBTYPE OF(number_space);
  END_ENTITY;
(*
```

5.2.5.10 number_space

A **number_space** is a **class_of_number** that is a continuum.

EXAMPLE The integers from 1 to 5 and the reals from 0.000 to 1.000 are examples of **number_space**.

EXPRESS specification:

```
*)
  ENTITY number_space
    SUPERTYPE OF (ONEOF(number_range, multidimensional_number_space))
    SUBTYPE OF(class_of_number);
  END_ENTITY;
(*
```

5.2.5.11 real_number

A **real_number** is an **arithmetic_number** that is a real number.

EXAMPLE 3.2146 is a representation of a **real_number**.

EXPRESS specification:

```
*)
  ENTITY real_number
    SUBTYPE OF(arithmetic_number);
  END_ENTITY;
(*
```

5.2.5.12 upper_bound_of_number_range

An **upper_bound_of_number_range** is a **relationship** that indicates an **arithmetic_number** is the largest value in a **number_range**.

EXAMPLE 5.3 is the upper bound of the range [3.1 to 5.3].

EXPRESS specification:

```
*)
  ENTITY upper_bound_of_number_range
    SUBTYPE OF(classification);
    SELF\classification.classified      :     arithmetic_number;
    SELF\classification.classifier      :     number_range;
  END_ENTITY;
(*
```

Attribute definitions:

classified : the **arithmetic_number** that as classified is the upper bound to the **number_range**

classifier : the **number_range** that as classifier is bounded in the **upper_bound_of_number_range**

5.2.6 Possible individuals

This subclause contains the declarations of entity data types that represent possible individuals.

NOTE Figure 182 is a diagram of the entity data type(s) defined in this subclause (see also 4.6.2 and 4.7).

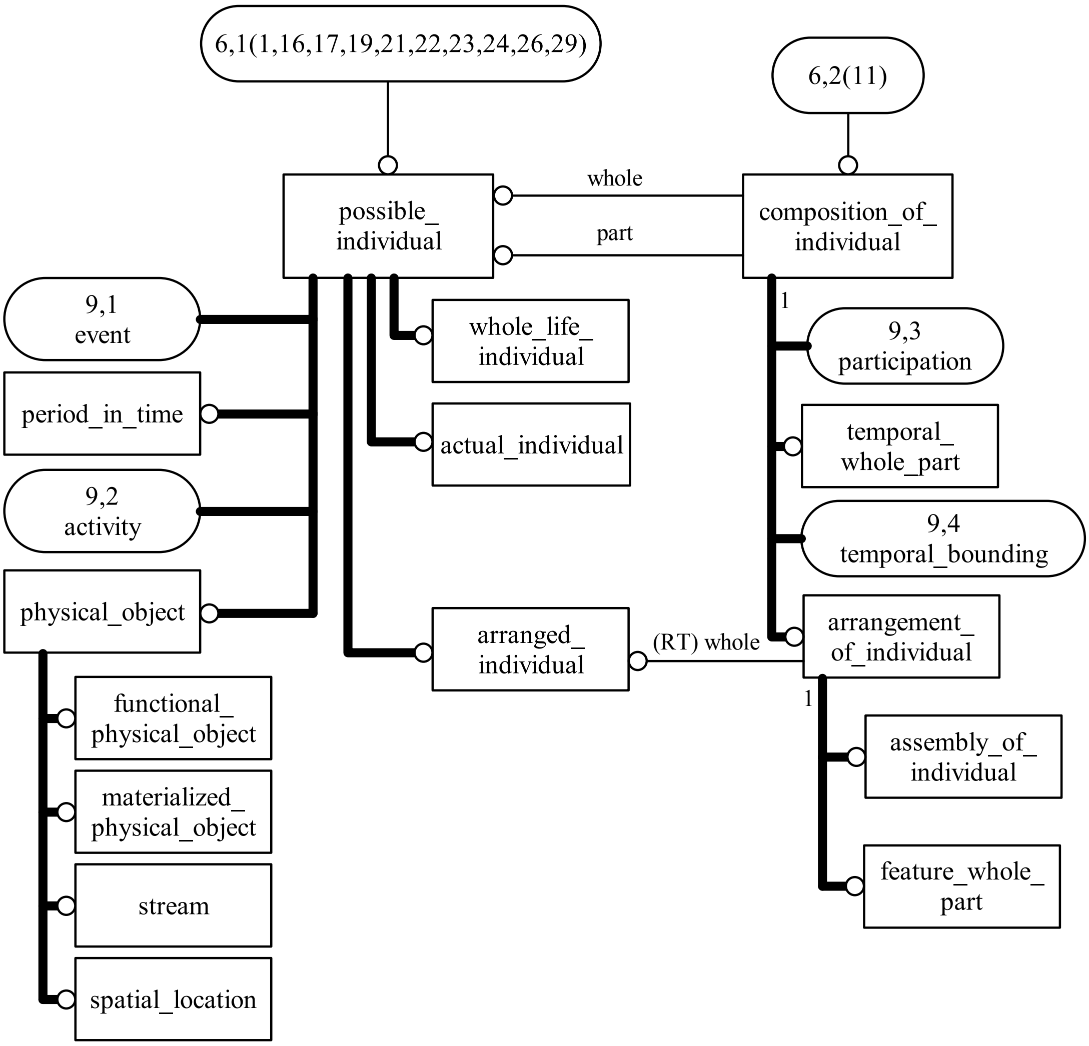


Figure 182 — lifecycle_integration_schema EXPRESS-G diagram 6 of 29

5.2.6.1 actual_individual

An **actual_individual** is a **possible_individual** that is a part of the space-time continuum that we inhabit. It exists in the present, past, or future of our universe, as opposed to some imagined universe.

NOTE The things we plan can usually only be assumed to be part of some imagined universe, until they come about.

EXAMPLE 1 The Eiffel Tower is an **actual_individual**.

EXAMPLE 2 The computer used to edit this part of ISO 15926 is an **actual_individual**.

EXAMPLE 3 The fictional character, Sherlock Holmes, is a **possible_individual** who is not an **actual_individual**.

EXAMPLE 4 The Earth in the year 2300 (assuming it still exists) is an **actual_individual**.

EXPRESS specification:

```
*)
  ENTITY actual_individual
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.6.2 arranged_individual

An **arranged_individual** is a **possible_individual** that has parts that play distinct roles with respect to the whole. The qualities of an **arranged_individual** are distinct from the qualities of its parts.

EXAMPLE 1 The vessel with serial number V-1234 is an **arranged_individual**.

EXAMPLE 2 The company Bloggs & Co. is an **arranged_individual**.

EXAMPLE 3 A laptop computer that consists of the main unit with its removable CD-ROM and floppy disk drives and power supply cables is an **arranged_individual**.

EXPRESS specification:

```
*)
  ENTITY arranged_individual
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.6.3 arrangement_of_individual

An **arrangement_of_individual** is a **composition_of_individual** that indicates that the part is a part of an **arranged_individual**. The temporal extent of the part is that of the whole.
An **arrangement_of_individual** may be an **assembly_of_individual**.

NOTE 1 The term "arranged" implies that parts have particular roles with respect to the whole.

NOTE 2 The natures of the relations to other parts of the whole are not specified by the arrangement relation. Relationships like **connection_of_individual** and **relative_location** would indicate this.

EXAMPLE 1 The relationship that indicates that a particular aircraft is flying as part of a formation can be represented by an instance of **arrangement_of_individual**.

EXAMPLE 2 The relationship that indicates that a particular bin in a warehouse is part of the warehouse layout can be represented by an instance of **arrangement_of_individual**.

EXPRESS specification:

```
*)
  ENTITY arrangement_of_individual
    SUPERTYPE OF (ONEOF(assembly_of_individual, feature_whole_part))
    SUBTYPE OF(composition_of_individual);
    SELF\composition_of_individual.whole     :     arranged_individual;
  END_ENTITY;
(*
```

Attribute definitions:

whole : the **arranged_individual** that is the whole in the **arrangement_of_individual**

5.2.6.4 assembly_of_individual

An **assembly_of_individual** is an **arrangement_of_individual** that indicates that the part is connected directly or indirectly to other parts of the whole. The parts and wholes are super-molecular objects.

NOTE Composition of molecules and smaller is represented through instances of **class_of_arrangement_of_individual**.

EXAMPLE The relation that indicates that a temporal part of an impeller is a part of an assembled pump can be represented by an instance of **assembly_of_individual**.

EXPRESS specification:

```
*)
  ENTITY assembly_of_individual
    SUBTYPE OF(arrangement_of_individual);
  END_ENTITY;
(*
```

5.2.6.5 composition_of_individual

A **composition_of_individual** is a **relationship** that indicates that the part **possible_individual** is a part of the whole **possible_individual**. A simple composition is indicated, unless a subtype is instantiated too. **composition_of_individual** is transitive.

NOTE Simple composition means that for example no arrangement of parts is necessarily implied or of concern. Where there is an arrangement of parts, this is indicated by an **arrangement_of_individual**, which, by being a subtype, implies also a simple composition.

EXAMPLE A grain of sand being part of a pile of sand is an example of **composition_of_individual**.

EXPRESS specification:

```
*)
  ENTITY composition_of_individual
    SUPERTYPE OF (ONEOF(arrangement_of_individual, temporal_whole_part,
                  participation, temporal_bounding))
    SUBTYPE OF(relationship);
    part                    :possible_individual;
    whole                   :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

part : the **possible_individual** that is part of the whole **possible_individual**

whole : the **possible_individual** that is the whole in the **composition_of_individual**

5.2.6.6 feature_whole_part

A **feature_whole_part** is an **arrangement_of_individual** that indicates that the part is a non-separable, contiguous part of the whole.

NOTE This includes wholes that cannot be non-destructively disassembled and reassembled such as the cast inlet flange of a pump.

EXAMPLE The relation that indicates that a flange face is part of a flange can be represented by an instance of **feature_whole_part**.

EXPRESS specification:

```
*)
  ENTITY feature_whole_part
    SUBTYPE OF(arrangement_of_individual);
  END_ENTITY;
(*
```

5.2.6.7 functional_physical_object

A **functional_physical_object** is a **physical_object** that has functional, rather than material, continuity as its basis for identity. Adjacent temporal parts of a **functional_physical_object** need not have common matter or energy, provided the matter or energy of each temporal part fulfils the same function.

EXAMPLE The heat exchanger system known as tag E-4507, which is part of a distillate transfer system, can be represented by an instance of **functional_physical_object**. This is distinct from the "shell and tube heat exchanger manufacture number ES/1234" that was installed as E-4507 when the plant was first built and later removed when worn out, to be replaced by a new heat exchanger with different serial number. "Shell and tube heat exchanger manufacture number ES/1234" and its differently numbered replacement can be represented by instances of **materialized_physical_object**. When ES/1234 is installed as E-4507 there is a temporal part of ES/1234 that is also a temporal part of E-4507.

EXPRESS specification:

```
*)
  ENTITY functional_physical_object
    SUBTYPE OF(physical_object);
  END_ENTITY;
(*
```

5.2.6.8 materialized_physical_object

A **materialized_physical_object** is a **physical_object** that has matter and/or energy continuity as its basis for identity. Matter or energy continuity requires some matter or energy to be common to adjacent temporal parts of the **materialized_physical_object**. Replacement of some components from time to time does not create a new identity.

EXAMPLE The shell and tube heat exchanger with manufacture's serial number ES/1234 can be represented by an instance of **materialized_physical_object**.

EXPRESS specification:

```
*)
  ENTITY materialized_physical_object
    SUBTYPE OF(physical_object);
  END_ENTITY;
(*
```

5.2.6.9 period_in_time

A **period_in_time** is a **possible_individual** that is all space for part of time - a temporal part of the universe.

EXAMPLE 1 July 2000 is an instance of **period_in_time**.

EXAMPLE 2 The period described by UTC 2000-11-21T06:00 to UTC 2000-11-21T11:53 is an instance of **period_in_time** compliant with ISO8601.

EXPRESS specification:

```
*)
  ENTITY period_in_time
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.6.10 physical_object

A **physical_object** is a **possible_individual** that is a distribution of matter, energy, or both.

EXAMPLE 1 A piece of metal is a **physical_object**.

EXAMPLE 2 A tree is a **physical_object**.

EXAMPLE 3 The thing identified by tag P101 is a **physical_object**.

EXAMPLE 4 A light beam is a **physical_object**.

EXAMPLE 5 A tank that is built and dismantled on site is both a **materialized_physical_object** and a **functional_physical_object**.

EXPRESS specification:

```
*)
  ENTITY physical_object
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.6.11 possible_individual

A **possible_individual** is a **thing** that exists in space and time. This includes:

— things where any of the space-time dimensions are vanishingly small,

— those that are either all space for any time, or all time and any space,

— the entirety of all space-time

— things that actually exist, or have existed,

— things that are fictional or conjectured and possibly exist in the past, present or future,

— temporal parts (states) of other individuals,

— things that have a specific position, but zero extent in one or more dimensions, such as points, lines, and surfaces.

In this context existence is based upon being imaginable within some consistent logic, including actual, hypothetical, planned, expected, or required individuals.

EXAMPLE The pump with serial number ABC123, Battersea Power Station, Sir Joseph Whitworth, Shakespeare, and the starship "Enterprise" can be represented by instances of **possible_individual**.

EXPRESS specification:

```
*)
  ENTITY possible_individual
    SUBTYPE OF(thing);
  END_ENTITY;
(*
```

5.2.6.12 spatial_location

A **spatial_location** is a **physical_object** that has continuity of relative position.

EXAMPLE Geographic datum, license block, construction area, country, air corridor, maritime traffic zone, hazard control zone, 4D points, lines, planes, solids.

EXPRESS specification:

```
*)
  ENTITY spatial_location
    SUBTYPE OF(physical_object);
  END_ENTITY;
(*
```

5.2.6.13 stream

A **stream** is a **physical_object** that is material or energy moving along a path, where the path is the basis of identity and may be constrained. The stream consists of the temporal parts of those things that are in the stream whilst they are in it.

EXAMPLE 1 Flux is a 4D-constrained case of **stream** where the path crosses a surface.

EXAMPLE 2 The naphtha flowing in a pipe between a crude distillation unit and a platformer is a **stream**.

EXPRESS specification:

```
*)
  ENTITY stream
    SUBTYPE OF(physical_object);
  END_ENTITY;
(*
```

5.2.6.14 temporal_whole_part

A **temporal_whole_part** is a **composition_of_individual** that indicates that one **possible_individual** is a temporal part of another **possible_individual**. The spatial extent of the temporal part is that of the temporal whole for the period of the existence of the temporal part.

Relationships that apply to the whole **possible_individual** also apply to the temporal parts of the **possible_individual**, except when the relationships relate to the temporal nature of the whole. So if a **possible_individual** is connected so are all its temporal parts, but being a **whole_life_individual** is not inherited by its temporal parts.

NOTE Since **temporal_whole_part** is transitive (inherited from its supertype) a hierarchy of temporal parts is possible, with a **whole_life_individual** at the top.

EXAMPLE 1 The relation that indicates that an operating period of a pump is a temporal part of the pump can be represented by an instance of **temporal_whole_part**.

EXAMPLE 2 The relationship that indicates that the time period known as March 1999 is part of the period known as 1st Quarter 1999 can be represented by an instance of **temporal_whole_part**.

EXPRESS specification:

```
*)
  ENTITY temporal_whole_part
    SUBTYPE OF(composition_of_individual);
  END_ENTITY;
(*
```

5.2.6.15 whole_life_individual

A **whole_life_individual** is a **possible_individual** that is a member of a **class_of_individual**, and is not a temporal part of any other **possible_individual** that is also a member of the same **class_of_individual**. A **whole_life_individual** includes its past and future.

NOTE A possible future temporal part of the **whole_life_individual** is a **possible_individual** that is related to the **whole_life_individual** by a **temporal_whole_part** relation.

EXAMPLE 1 A plastic cup (bounded by its creation and destruction **event**s) can be represented by an instance of **whole_life_individual**. The cup whilst it stands on this table is a temporal part of this **whole_life_individual**.

EXAMPLE 2 The universe for all time is a **whole_life_individual**.

EXPRESS specification:

```
*)
  ENTITY whole_life_individual
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.7 Classes of individual

This subclause contains the declarations of entity data types that represent classes of individual.

NOTE Figure 183 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4 and 4.8.4.10).

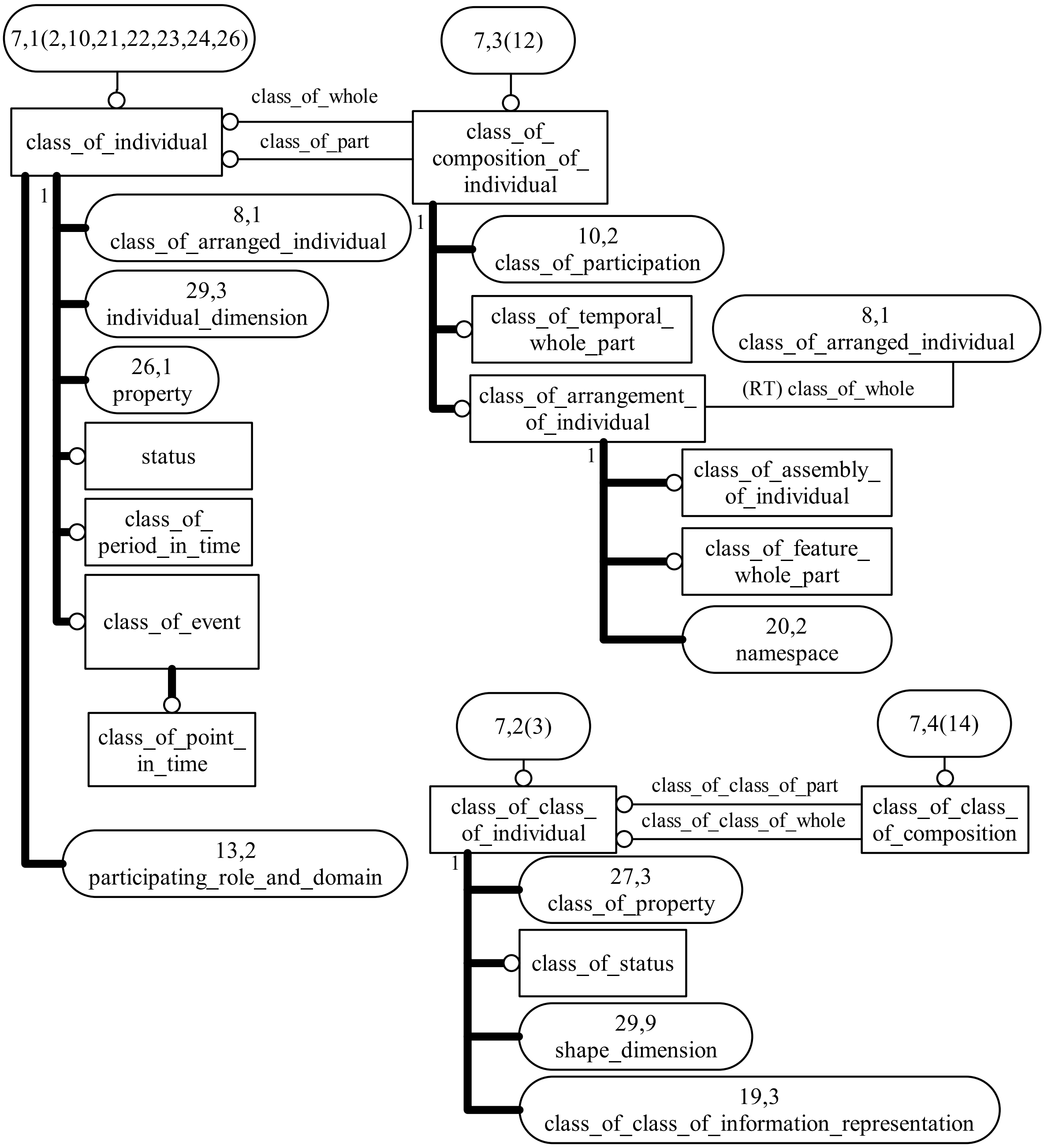

7,1(2,10,21,22,23,24,26)
7,3(12)
class_of_whole
class_of_part
class_of_individual
class_of_ composition_of_ individual
1
8,1 class_of_arranged_individual
29,3 individual_dimension
26,1 property
status
class_of_ period_in_time
class_of_event
class_of_point_ in_time
13,2 participating_role_and_domain
10,2 class_of_participation
class_of_temporal_ whole_part
class_of_arrangement_ of_individual
8,1 class_of_arranged_individual
(RT) class_of_whole
class_of_assembly_ of_individual
class_of_feature_ whole_part
20,2 namespace
7,2(3)
7,4(14)
class_of_class_of_part
class_of_class_of_whole
class_of_class_ of_individual
class_of_class_ of_composition
27,3 class_of_property
class_of_status
29,9 shape_dimension
19,3 class_of_class_of_information_representation

Figure 183 — lifecycle_integration_schema EXPRESS-G diagram 7 of 29

5.2.7.1 class_of_arrangement_of_individual

A **class_of_arrangement_of_individual** is a **class_of_composition_of_individual** whose members are instances of **arrangement_of_individual**.

EXAMPLE The fact that water is made up of H_2O molecules is an instance of **class_of_arrangement_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_arrangement_of_individual
```

```
    SUPERTYPE OF (ONEOF(class_of_feature_whole_part,
                        class_of_assembly_of_individual, namespace))
    SUBTYPE OF(class_of_composition_of_individual);
    SELF\class_of_composition_of_individual.class_of_whole     :
                                    class_of_arranged_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_whole : the **class_of_arranged_individual** that is the **class_of_whole** in the **class_of_arrangement_of_individual**

5.2.7.2 class_of_assembly_of_individual

A **class_of_assembly_of_individual** is a **class_of_arrangement_of_individual** whose members are instances of **assembly_of_individual**.

EXAMPLE That impellers are parts of centrifugal pumps is a **class_of_assembly_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_assembly_of_individual
    SUBTYPE OF(class_of_arrangement_of_individual);
  END_ENTITY;
(*
```

5.2.7.3 class_of_class_of_composition

A **class_of_class_of_composition** is a **class_of_class_of_relationship** whose members are instances of **class_of_composition**. It indicates that a member of a member of the class_of_class_of_part is a part of a member of an instance of the class_of_class_of_whole.

EXAMPLE Toxicity description is a class_of_class_of_part of a material data sheet, where the description "has carcinogenic components" is a class_of_part on the Mogas Material Safety Data Sheet, and copy #5 of the Mogas Material Safety Data Sheet has "has carcinogenic components" as a part.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_composition
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_part     :class_of_class_of_individual;
    class_of_class_of_whole    :class_of_class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_class_of_part : the **class_of_class_of_individual** that is the **class_of_class_of_part** in the **class_of_class_of_composition**

class_of_class_of_whole : the **class_of_class_of_individual** that is the **class_of_class_of_whole** in the **class_of_class_of_composition**

5.2.7.4 class_of_class_of_individual

A **class_of_class_of_individual** is a **class_of_class** whose members are instances of **class_of_individual**.

EXAMPLE "Premium Product" is a **class_of_class_of_individual** that has **class_of_individual** "mogas" as a member.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_individual
    SUPERTYPE OF (ONEOF(class_of_class_of_information_representation,
                        class_of_property, class_of_status,
                        shape_dimension))
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.7.5 class_of_composition_of_individual

A **class_of_composition_of_individual** is a **class_of_relationship** whose members are members of **composition_of_individual**.

EXAMPLE That piles of sand may have grains of sand as parts is an example of **class_of_composition_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_composition_of_individual
    SUPERTYPE OF (ONEOF(class_of_arrangement_of_individual,
                        class_of_temporal_whole_part,
                        class_of_participation))
    SUBTYPE OF(class_of_relationship);
    class_of_part               :class_of_individual;
    class_of_whole              :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_part : the **class_of_individual** that is the **class_of_part** in the **class_of_composition_of_individual**

class_of_whole : the **class_of_individual** that is the **class_of_whole** in the **class_of_composition_of_individual**

5.2.7.6 class_of_event

A **class_of_event** is a **class_of_individual** whose members are members of **event**.

EXAMPLE Continuous and instantaneous are instances of **class_of_event**. A continuous event is one such as a stream boundary flowing through a pipe.

EXPRESS specification:

```
*)
  ENTITY class_of_event
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.7.7 class_of_feature_whole_part

A **class_of_feature_whole_part** is a **class_of_arrangement_of_individual** whose members are instances of **feature_whole_part**.

EXAMPLE Thermowells have stems, and tables have tops are examples of **class_of_feature_whole_part**.

EXPRESS specification:

```
*)
  ENTITY class_of_feature_whole_part
    SUBTYPE OF(class_of_arrangement_of_individual);
  END_ENTITY;
(*
```

5.2.7.8 class_of_individual

A **class_of_individual** is a class whose members are instances of **possible_individual**.

EXAMPLE The class known as 'engineer', whose members are people qualified or skilled in engineering principles and practices can be represented by an instance of **class_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_individual
    SUPERTYPE OF (ONEOF(
                      class_of_event,
                      class_of_arranged_individual,
                      class_of_period_in_time,
                      individual_dimension,
                      property,
                      status
                      )
                      ANDOR participating_role_and_domain)
    SUBTYPE OF(class);
  END_ENTITY;
(*
```

5.2.7.9 class_of_period_in_time

A **class_of_period_in_time** is a **class_of_individual** whose members are instances of **period_in_time**.

EXAMPLE Monday and June are examples of **class_of_period_in_time**.

EXPRESS specification:

```
*)
  ENTITY class_of_period_in_time
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.7.10 class_of_point_in_time

A **class_of_point_in_time** is a **class_of_event** whose members are members of **point_in_time**.

EXAMPLE Midnight is a **class_of_point_in_time.**

EXPRESS specification:

```
*)
  ENTITY class_of_point_in_time
    SUBTYPE OF(class_of_event);
  END_ENTITY;
(*
```

5.2.7.11 class_of_status

A **class_of_status** is a **class_of_class_of_individual** whose members are a **status**.

EXAMPLE An example of **class_of_status** is approval, with members: not assessed, approved, rejected.

EXPRESS specification:

```
*)
  ENTITY class_of_status
    SUBTYPE OF(class_of_class_of_individual);
  END_ENTITY;
(*
```

5.2.7.12 class_of_temporal_whole_part

A **class_of_temporal_whole_part** is a **class_of_composition_of_individual** whose members are members of **temporal_whole_part**.

EXAMPLE The class that indicates that Crude Distillation Units may have a maximum naphtha mode can be represented by an instance of **class_of_temporal_whole_part**.

EXPRESS specification:

```
*)
  ENTITY class_of_temporal_whole_part
    SUBTYPE OF(class_of_composition_of_individual);
  END_ENTITY;
(*
```

5.2.7.13 status

A **status** is a **class_of_individual** that is a characteristic or quality that is described by discrete, unordered values.

EXAMPLE The classes known as 'open', 'painted', 'approved', 'old', 'new', 'worn', 'hazardous', 'safe', 'dangerous', 'happy', 'sad', and 'rusty' can all be represented as instances of **status**.

NOTE Degrees of openness or paintedness are represented as instances of **property** and not instances of **status**.

EXPRESS specification:

```
*)
  ENTITY status
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.8 Classes of arranged individual

This subclause contains the declarations of entity data types that represent classes of arranged individual.

NOTE Figure 184 is a diagram of the entity data type(s) defined in this subclause (see also 4.7.9 and 4.8.4.1).

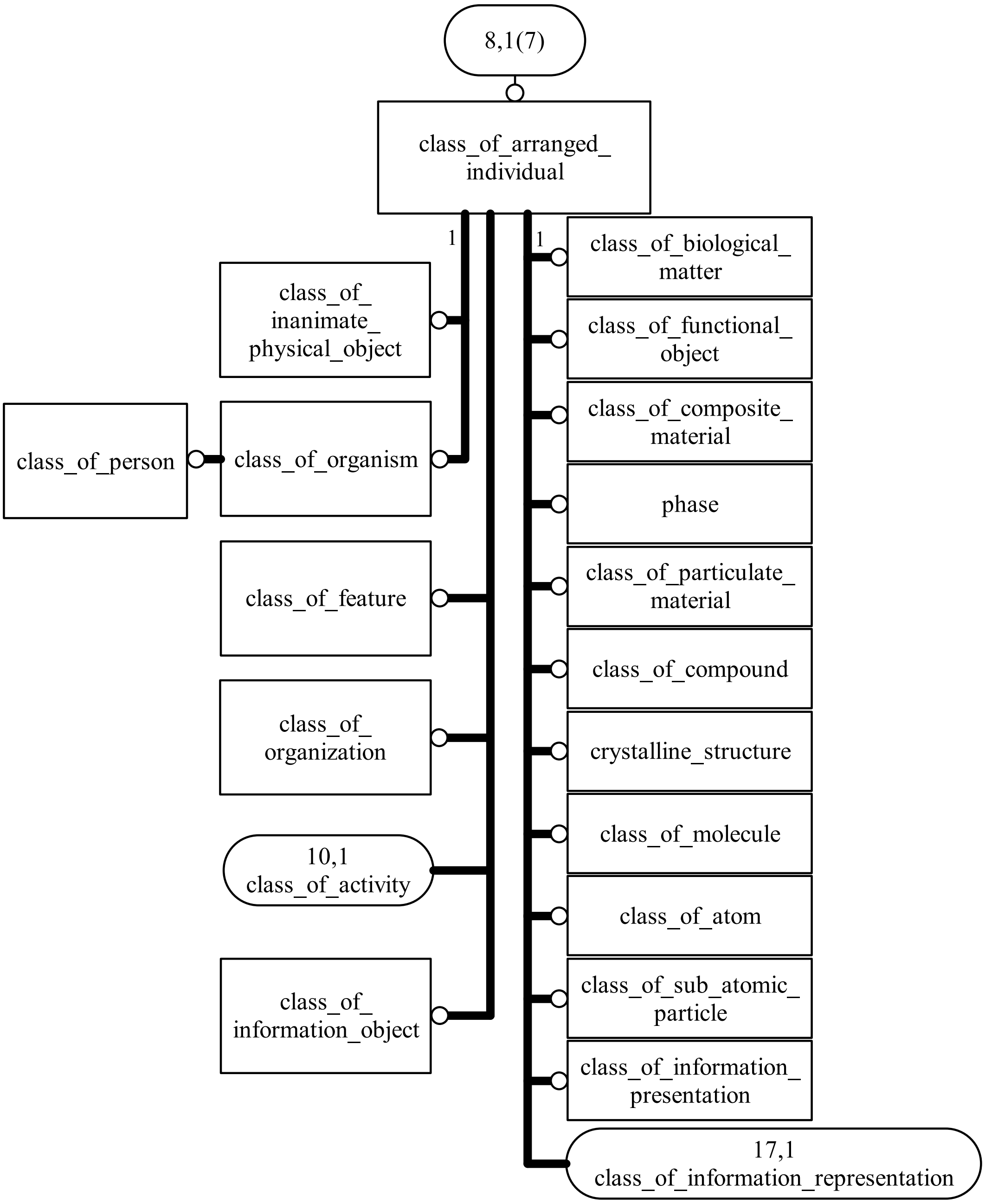

8,1(7)
class_of_arranged_individual
1
1
class_of_inanimate_physical_object
class_of_person
class_of_organism
class_of_feature
class_of_organization
10,1 class_of_activity
class_of_information_object
class_of_biological_matter
class_of_functional_object
class_of_composite_material
phase
class_of_particulate_material
class_of_compound
crystalline_structure
class_of_molecule
class_of_atom
class_of_sub_atomic_particle
class_of_information_presentation
17,1 class_of_information_representation

Figure 184 — lifecycle_integration_schema EXPRESS-G diagram 8 of 29

5.2.8.1 class_of_arranged_individual

A **class_of_arranged_individual** is a **class_of_individual** whose members are an arrangement of components.

EXAMPLE Robocop is a **class_of_arranged_individual** that has some parts that are members of some **class_of_inanimate_physical_object** and parts that are members of some **class_of_organism**.

NOTE 1 The EXPRESS ONEOF (mutually exclusive) constraint on some of the subtypes does not prevent a particular **possible_individual** from being, say, a member of a particular **arranged_individual** classified by **class_of_biological_matter** and a member of a particular **class_of_composite_material**. It is only the classes themselves that are not members of more than one of the entity types.

NOTE 2 Specifications or descriptions of useful objects are often intersections of several arrangement classes, allowing both shape and material aspects to be constrained. In this part of ISO 15926, such intersections are members of **class_of_arranged_individual**, **class_of_feature**, **class_of_inanimate_physical_object**, **class_of_organization**, **class_of_activity**, **class_of_organism**, or **class_of_information_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_arranged_individual
    SUPERTYPE OF (ONEOF(
                        class_of_atom,
                        class_of_biological_matter,
                        class_of_composite_material,
                        class_of_compound,
                        class_of_functional_object,
                        class_of_information_presentation,
                        class_of_information_representation,
                        class_of_molecule,
                        class_of_particulate_material,
                        class_of_sub_atomic_particle,
                        crystalline_structure,
                        phase)
                        ANDOR class_of_organization
                        ANDOR class_of_activity
                        ANDOR class_of_information_object
                        ANDOR class_of_feature
                        ANDOR ONEOF(class_of_organism,
                        class_of_inanimate_physical_object))
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.8.2 class_of_atom

A **class_of_atom** is a **class_of_arranged_individual** whose members are atoms.

EXAMPLE All entries in the periodic table of elements can be represented by instances of **class_of_atom**.

EXPRESS specification:

```
*)
  ENTITY class_of_atom
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.3 class_of_biological_matter

A **class_of_biological_matter** is a **class_of_arranged_individual** whose members are particular types of cell or aggregations of cells.

EXAMPLE The classes known as 'blood', 'enzyme', and 'plasma' can be represented by instances of **class_of_biological_matter**.

EXPRESS specification:

```
*)
  ENTITY class_of_biological_matter
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.4 class_of_composite_material

A **class_of_composite_material** is a **class_of_arranged_individual** whose members have a common arrangement of separable compounds.

EXAMPLE 1 Laminates such as plywood, fibreglass, and carbon fibre can be represented by instances of **class_of_composite_material**.

EXAMPLE 2 Wood, muscle, and skin can be represented by instances of **class_of_composite_material**.

EXPRESS specification:

```
*)
  ENTITY class_of_composite_material
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.5 class_of_compound

A **class_of_compound** is a **class_of_arranged_individual** whose members consist of arrangements of molecules of the same or different types, bound together by intermolecular forces. This includes both mixtures and alloys.

EXAMPLE Water, sulphuric acid, sand, limestone, and steel can be represented by instances of **class_of_compound**.

EXPRESS specification:

```
*)
  ENTITY class_of_compound
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.6 class_of_feature

A **class_of_feature** is a **class_of_arranged_individual** whose members are contiguous, non-separable parts of some **possible_individual** and have an incompletely defined boundary.

EXAMPLE The classes known as 'mountain', 'groove', 'rim', 'nozzle', 'nose', and 'raised face' can all be represented as instances of **class_of_feature**.

EXPRESS specification:

```
*)
  ENTITY class_of_feature
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.7 class_of_functional_object

A **class_of_functional_object** is a **class_of_arranged_individual** that indicates the function or purpose of an object.

EXAMPLE Pump, valve, and car are examples of **class_of_functional_object**. Particular models of pump, valve, car, etc are instances of **class_of_inanimate_physical_object** that are specializations of these instances of **class_of_functional_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_functional_object
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.8 class_of_inanimate_physical_object

A **class_of_inanimate_physical_object** is a **class_of_arranged_individual** whose members are not living.

EXAMPLE The class known as 'oil' can be represented by an instance of **class_of_inanimate_physical_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_inanimate_physical_object
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.9 class_of_information_object

A **class_of_information_object** is a **class_of_arranged_individual** whose members are members of zero or more **class_of_information_representation** and of zero or more **class_of_information_presentation**.

NOTE Usually, it is a **physical_object** (like a paper document) that is classified as a **class_of_information_object**.

EXAMPLE ‘Newspaper’ is a **class_of_information_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_information_object
```

```
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.10 class_of_information_presentation

A **class_of_information_presentation** is a **class_of_arranged_individual** that distinguishes styles for presenting information.

EXAMPLE The character styles bold, italic, Times New Roman, and 16pt can be represented as instances of **class_of_information_presentation**.

EXPRESS specification:

```
*)
  ENTITY class_of_information_presentation
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.11 class_of_molecule

A **class_of_molecule** is a **class_of_arranged_individual** whose members are molecules.

EXAMPLE H_2O, H_2SO_4, and DNA can be represented by instances of **class_of_molecule**.

EXPRESS specification:

```
*)
  ENTITY class_of_molecule
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.12 class_of_organism

A **class_of_organism** is a **class_of_arranged_individual** whose members are living organisms.

EXAMPLE Human being, sheep, earthworm, oak tree, and bacteria are instances of **class_of_organism**.

EXPRESS specification:

```
*)
  ENTITY class_of_organism
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.13 class_of_organization

A **class_of_organization** is a **class_of_arranged_individual** whose members are instances of **physical_object** that are composed of temporal parts of people and other assets, and are organised with a particular purpose.

EXAMPLE A company, government, or project team can be represented by instances of **class_of_organization**.

EXPRESS specification:

```
*)
  ENTITY class_of_organization
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.14 class_of_particulate_material

A **class_of_particulate_material** is a **class_of_arranged_individual** whose members are arranged amounts of super-molecular sized objects of the same or different types.

EXAMPLE Pile of sand, sand and cement mix, bag of bolts, catalyst fill for a reactor are examples of **class_of_particulate_material**.

EXPRESS specification:

```
*)
  ENTITY class_of_particulate_material
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.15 class_of_person

A **class_of_person** is a **class_of_organism** whose members are people.

EXAMPLE An engineer, plant manager, student, male, female, senior citizen, adult, girl, or boy can be represented by instances of **class_of_person**. An engineer, plant manager, or student is also an instance of **class_of_functional_object**.

EXPRESS specification:

```
*)
  ENTITY class_of_person
    SUBTYPE OF(class_of_organism);
  END_ENTITY;
(*
```

5.2.8.16 class_of_sub_atomic_particle

A **class_of_sub_atomic_particle** is a **class_of_arranged_individual** whose members are constituent particles of atoms.

EXAMPLE Proton, electron, meson, neutron, positron, muon, quark, and neutrino can be represented by instances of **class_of_sub_atomic_particle**.

EXPRESS specification:

```
*)
  ENTITY class_of_sub_atomic_particle
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.17 crystalline_structure

A **crystalline_structure** is a **class_of_arranged_individual** that is a form in which many simple elements and their natural compounds regularly aggregate by the operation of natural affinity: it has a definite internal structure, with the external form of a solid enclosed by a number of symmetrically arranged plane faces, and varying in simplicity from the cube to much more complicated geometric bodies.

EXAMPLE Ferritic, martensitic, and austenitic are examples of **crystalline_structure**.

EXPRESS specification:

```
*)
  ENTITY crystalline_structure
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.8.18 phase

A **phase** is a **class_of_arranged_individual** based on the nature of the boundary behaviour of material resulting from its atomic and molecular bonding.

NOTE **phase** excludes types of internal structure such as crystalline.

EXAMPLE The classes known as 'liquid' and 'solid' can be represented by instances of **phase**.

EXPRESS specification:

```
*)
  ENTITY phase
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.9 Activities and events

This subclause contains the declarations of entity data types that represent activities and events.

NOTE Figure 185 is a diagram of the entity data type(s) defined in this subclause (see also 4.7.10 and 4.7.17).

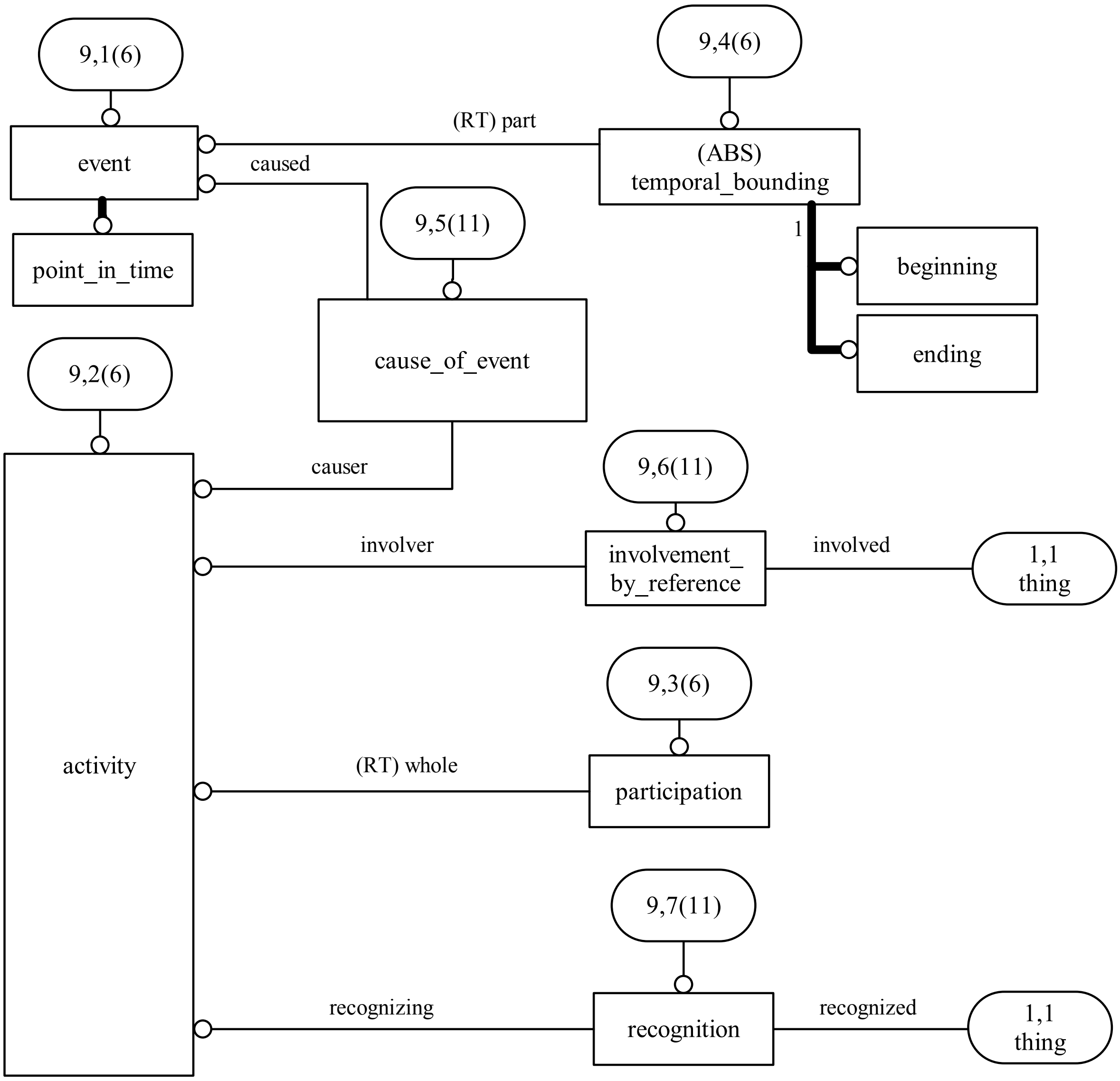


Figure 185 — lifecycle_integration_schema EXPRESS-G diagram 9 of 29

5.2.9.1 activity

An **activity** is a **possible_individual** that brings about change by causing the **event** that marks the **beginning**, or the **event** that marks the **ending** of a **possible_individual**.

An activity consists of the temporal parts of those members of **possible_individual** that participate in the activity. The participating temporal parts will be classified by the **participating_role_and_domain** that indicates the role of the temporal part in the **activity**.

EXAMPLE Pumping a fluid with a mechanical pump can be represented by an instance of **activity**.

EXPRESS specification:

```
*)
  ENTITY activity
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.9.2 beginning

A **beginning** is a **temporal_bounding** that marks the temporal start of a **possible_individual**.

EXAMPLE 1 The relation that indicates that the **point_in_time** known as 0000hrs 1st July 1999 UTC is the beginning of the **period_in_time** known as July 1999 UTC can be represented by an instance of **beginning**.

EXAMPLE 2 The relation that indicates that the **event** 'loading complete' marks the start of the **possible_individual** 'loading plant idle' can be represented by an instance of **beginning**.

EXPRESS specification:

```
*)
  ENTITY beginning
    SUBTYPE OF(temporal_bounding);
  END_ENTITY;
(*
```

5.2.9.3 cause_of_event

A **cause_of_event** is a **relationship** that indicates that the caused **event** is caused by the causer **activity**.

EXAMPLE The relation that indicates that the tanker loading activity caused the **event** described as 'tank liquid level full' can be represented by an instance of **cause_of_event**.

EXPRESS specification:

```
*)
  ENTITY cause_of_event
    SUBTYPE OF(relationship);
    caused                      :event;
    causer                      :activity;
  END_ENTITY;
(*
```

Attribute definitions:

caused : the **event** that is caused in the **cause_of_event**

causer : the **activity** that is the causer in the **cause_of_event**

5.2.9.4 ending

An **ending** is a **temporal_bounding** that marks the end of a **possible_individual**.

EXAMPLE 1 The relation that indicates that the **point_in_time** known as 0000hrs 1st July 1999 GMT is the end of the **period_in_time** known as June 1999 GMT can be represented by an instance of **ending**.

EXAMPLE 2 The relation that indicates that the **event** 'loading complete' marks the end of the **possible_individual** 'loading plant operating period 1' (a temporal part of the loading plant) is an instance of **ending**.

EXPRESS specification:

```
*)
  ENTITY ending
    SUBTYPE OF(temporal_bounding);
  END_ENTITY;
(*
```

5.2.9.5 event

An **event** is a **possible_individual** with zero extent in time. An **event** is the temporal boundary of one or more **possible_individual**s, although there may be no knowledge of these **possible_individual**s.

EXAMPLE The connection of power to a pump is an event that marks the beginning of a temporal part of that pump.

EXPRESS specification:

```
*)
  ENTITY event
    SUBTYPE OF(possible_individual);
  END_ENTITY;
(*
```

5.2.9.6 involvement_by_reference

An **involvement_by_reference** is a **relationship** that indicates that a **thing** is referred to in an **activity**.

NOTE This entity type is for involvements that are not direct **participation** of a **possible_individual**, such as involvement of a class, or of a historical or future temporal part of a **possible_individual**.

EXAMPLE A conversation that refers to the Roman Empire is an **activity** that relates to the Roman Empire by an **involvement_by_reference**.

EXPRESS specification:

```
*)
  ENTITY involvement_by_reference
    SUBTYPE OF(relationship);
    involved                :thing;
    involver                :activity;
  END_ENTITY;
(*
```

Attribute definitions:

involved : the **thing** that is involved in the referenced **activity**

involver : the **activity** in which the referenced **thing** is involved

5.2.9.7 participation

A **participation** is a **composition_of_individual** that indicates that a **possible_individual** is a participant in an **activity**.

NOTE The **possible_individual** that is the part in the **participation** is may be a temporal part of a **whole_life_individual** that is classified by the **role_and_domain** that indicates the role it plays in the **activity**.

EXAMPLE The relationship between the temporal part of P1234 that performs the discharge of the Motor Vessel Murex on 2nd December 2002, and the activity that is that discharge of that vessel is a **participation**.

EXPRESS specification:

```
*)
  ENTITY participation
    SUBTYPE OF(composition_of_individual);
    SELF\composition_of_individual.whole      :     activity;
  END_ENTITY;
(*
```

Attribute definitions:

whole : the **activity** that is the whole in the **participation**

5.2.9.8 point_in_time

A **point_in_time** is an **event** that is the whole space extension with zero extent in time.

NOTE In using this part of ISO15926, a **point_in_time** should be represented by a **representation_of_Gregorian_date_and_UTC_time**.

EXAMPLE The time known as UTC 1999-05-13T16:31:23.56 is a **point_in_time**.

EXPRESS specification:

```
*)
  ENTITY point_in_time
    SUBTYPE OF(event);
  END_ENTITY;
(*
```

5.2.9.9 recognition

A **recognition** is a **relationship** that indicates that a **thing** is recognized through an **activity**.

EXAMPLE Measurement activity #358 recognized that the room was a member of the 20 Celsius **property**.

EXPRESS specification:

```
*)
  ENTITY recognition
    SUBTYPE OF(relationship);
    recognized                :thing;
    recognizing               :activity;
  END_ENTITY;
(*
```

Attribute definitions:

recognized : the **thing** that is recognized by the **activity**

recognizing : the **activity** that results in the recognition

5.2.9.10 temporal_bounding

A **temporal_bounding** is an **assembly_of_individual** that indicates that the part **event** is a temporal boundary of the whole **possible_individual**.

EXPRESS specification:

```
*)
  ENTITY temporal_bounding
    ABSTRACT SUPERTYPE OF (ONEOF(ending, beginning))
    SUBTYPE OF(composition_of_individual);
    SELF\composition_of_individual.part :     event;
  END_ENTITY;
(*
```

Attribute definitions:

part : the **event** that is the part in the **temporal_bounding**

5.2.10 Classes of activity

This subclause contains the declarations of entity data types that represent classes of activity.

NOTE Figure 186 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4.9).

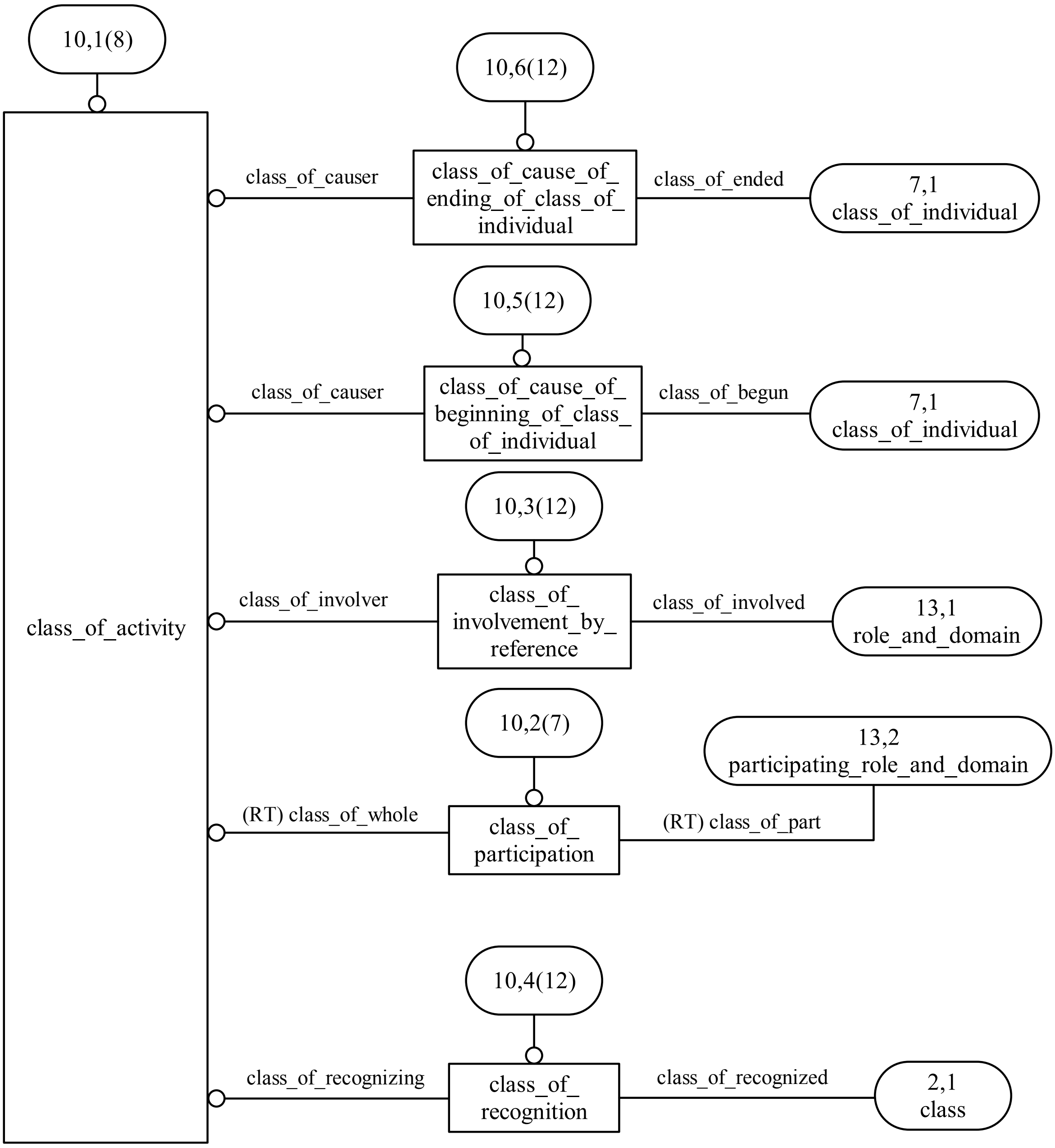


Figure 186 — lifecycle_integration_schema EXPRESS-G diagram 10 of 29

5.2.10.1 class_of_activity

A **class_of_activity** is a **class_of_arranged_individual** whose members are instances of **activity**.

EXAMPLE Drilling, distilling, and approving can be represented by instances of **class_of_activity**.

NOTE Behaviour is a term used to describe a **class_of_activity** either where there are preconditions and the **class_of_activity** is a response to those preconditions, e.g. reaction to touching a hot surface, or where the way an activity occurs is described by some property or function, e.g. fluid flow being described by the viscosity of the fluid.

EXPRESS specification:

```
*)
  ENTITY class_of_activity
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;
(*
```

5.2.10.2 class_of_cause_of_beginning_of_class_of_individual

A **class_of_cause_of_beginning_of_class_of_individual** is a **class_of_relationship** that indicates that a member of a **class_of_activity** causes the beginning of a member of a **class_of_individual**.

EXAMPLE A car manufacturing activity causes the beginning of a car.

EXPRESS specification:

```
*)
  ENTITY class_of_cause_of_beginning_of_class_of_individual
    SUBTYPE OF(class_of_relationship);
    class_of_begun            :class_of_individual;
    class_of_causer           :class_of_activity;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_begun : the **class_of_individual** a member of which is created by a member of the **class_of_activity**

class_of_causer : the **class_of_activity** whose members cause a member of the **class_of_individual** to begin

5.2.10.3 class_of_cause_of_ending_of_class_of_individual

A **class_of_cause_of_ending_of_class_of_individual** is a **class_of_relationship** that indicates that a member of the **class_of_activity** causes the ending of a member of the **class_of_individual**.

EXAMPLE A car crushing activity causes the end of the life of a car.

EXPRESS specification:

```
*)
  ENTITY class_of_cause_of_ending_of_class_of_individual
    SUBTYPE OF(class_of_relationship);
    class_of_causer           :class_of_activity;
    class_of_ended            :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_causer : the **class_of_activity** that a member of which causes the end of life of a member of the **class_of_individual**

class_of_ended : the **class_of_individual** a member of which is ended by a member of the **class_of_activity**

5.2.10.4 class_of_involvement_by_reference

A **class_of_involvement_by_reference** is a **class_of_relationship** whose members are instances of **involvement_by_reference**.

EXAMPLE Discussion of historical activities is an example of **class_of_involvement_by_reference**.

EXPRESS specification:

```
*)
  ENTITY class_of_involvement_by_reference
    SUBTYPE OF(class_of_relationship);
    class_of_involved          :role_and_domain;
    class_of_involver          :class_of_activity;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_involved : the **role_and_domain** that has the **class_of_involvement_by_reference**

class_of_involver : the **class_of_activity** that has the **class_of_involvement_by_reference**

5.2.10.5 class_of_participation

A **class_of_participation** is a **class_of_composition_of_individual** that indicates a member of an instance of **participating_role_and_domain** participates in a member of an instance of **class_of_activity**.

EXAMPLE "Conductor of a musical performance" is an example of **class_of_participation**.

EXPRESS specification:

```
*)
  ENTITY class_of_participation
    SUBTYPE OF(class_of_composition_of_individual);
    SELF\class_of_composition_of_individual.class_of_part      :
                                  participating_role_and_domain;
    SELF\class_of_composition_of_individual.class_of_whole     :
                                  class_of_activity;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_part : the **participating_role_and_domain** that has the **class_of_participation**

class_of_whole : the **class_of_activity** that has the **class_of_participation**

5.2.10.6 class_of_recognition

A **class_of_recognition** is a **class_of_relationship** that indicates that a member of a **class_of_activity** may result in the recognition of a member of a **class**.

EXAMPLE A measurement activity may result in the recognition of the **classification** of a **possible_individual** by a **property**.

EXPRESS specification:

```
*)
  ENTITY class_of_recognition
    SUBTYPE OF(class_of_relationship);
    class_of_recognized        :class;
    class_of_recognizing       :class_of_activity;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_recognized : the **class** whose members are recognized by members of the **class_of_activity**

class_of_recognizing : the **class_of_activity** whose members perform the recognition of the **class**

5.2.11 Relationships

This subclause contains the declarations of entity data types that represent relationships.

NOTE Figure 187 is a diagram of the entity data type(s) defined in this subclause (see also 4.6.4 and 4.10.1).

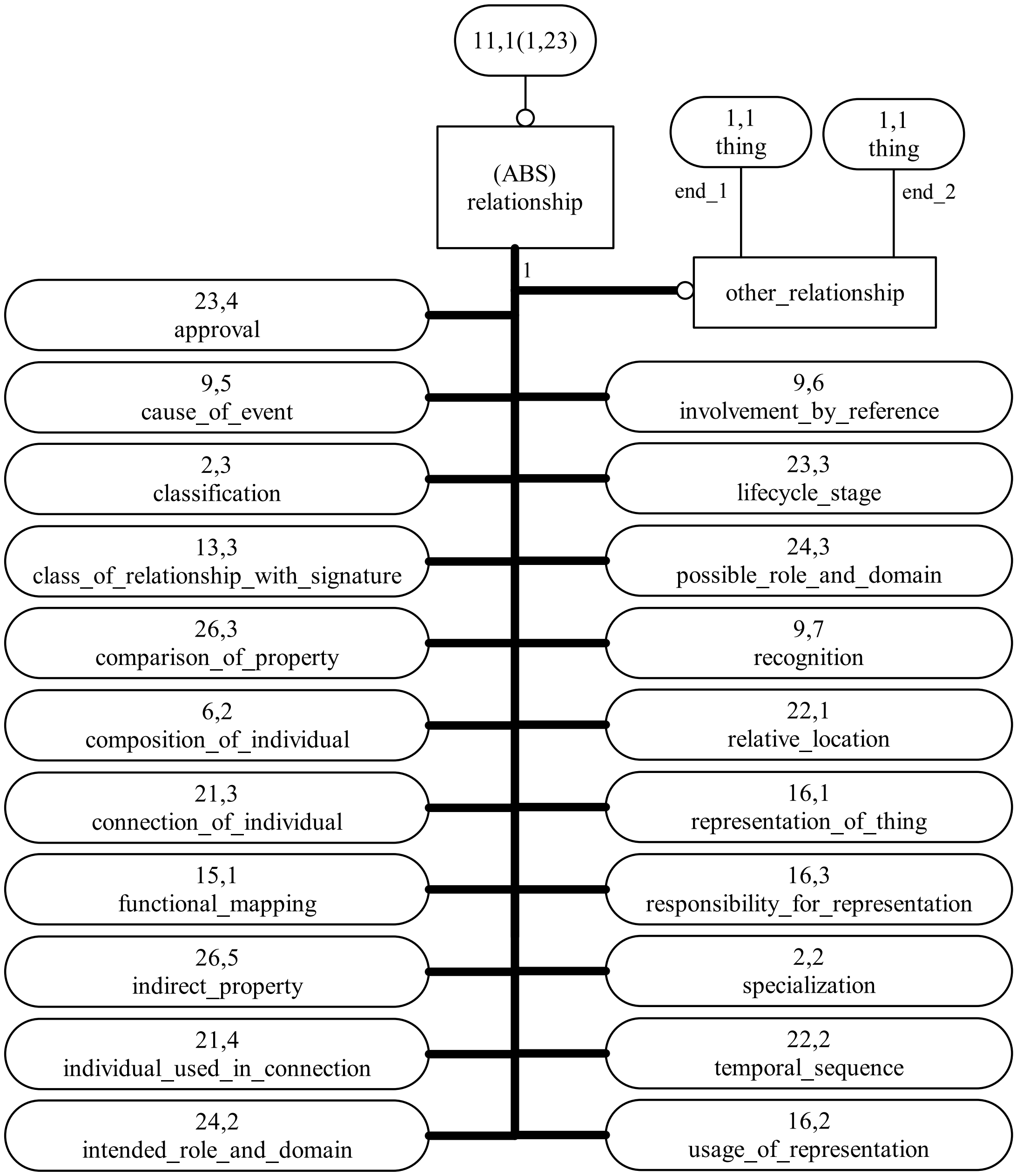


Figure 187 — lifecycle_integration_schema EXPRESS-G diagram 11 of 29

5.2.11.1 other_relationship

An **other_relationship** is a **relationship** that is not a member of any of the other explicit subtypes of **relationship**. The meaning of an **other_relationship** is specified by a **classification** by an instance of **class_of_relationship_with_signature**.

EXAMPLE The **relationship** that indicates that a car is manufactured by Ford can be represented by an instance of **other_relationship**.

The **role_and_domain** that classifies the **end_1** and **end_2** attributes is given by the **class_of_end_1** and **class_of_end_2** attributes respectively for the **class_of_relationship_with_signature** that classifies the **other_relationship**. Where the **class_of_relationship_with_signature** is also a

class_of_relationship_with_related_end_1 or a **class_of_relationship_with_related_end_2** then the **end_1** or **end_2** respectively of the **other_relationship** shall take the value specified by the related attribute.

EXPRESS specification:

```
*)
  ENTITY other_relationship
    SUBTYPE OF(relationship);
    end_1                      :thing;
    end_2                      :thing;
  END_ENTITY;
(*
```

Attribute definitions:

end_1 : the first of two instances of **thing** that are related

end_2 : the second of two instances of **thing** that are related

5.2.11.2 relationship

A **relationship** is an **abstract_object** that indicates something that one thing has to do with another.

NOTE Only classes of binary relationship are supported. More complex objects can be supported using **multidimensional_object** and **class_of_multidimensional_object**.

EXPRESS specification:

```
*)
  ENTITY relationship
    ABSTRACT SUPERTYPE OF (ONEOF(
                        approval,
                        cause_of_event,
                        class_of_relationship_with_signature,
                        classification,
                        comparison_of_property,
                        composition_of_individual,
                        connection_of_individual,
                        functional_mapping,
                        indirect_property,
                        individual_used_in_connection,
                        intended_role_and_domain,
                        involvement_by_reference,
                        lifecycle_stage,
                        other_relationship,
                        possible_role_and_domain,
                        recognition,
                        relative_location,
                        representation_of_thing,
                        responsibility_for_representation,
                        specialization,
                        temporal_sequence,
                        usage_of_representation))
    SUBTYPE OF(abstract_object);
  END_ENTITY;
(*
```

5.2.12 Classes of relationship

This subclause contains the declarations of entity data types that represent classes of relationship.

NOTE Figure 188 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.3.3).

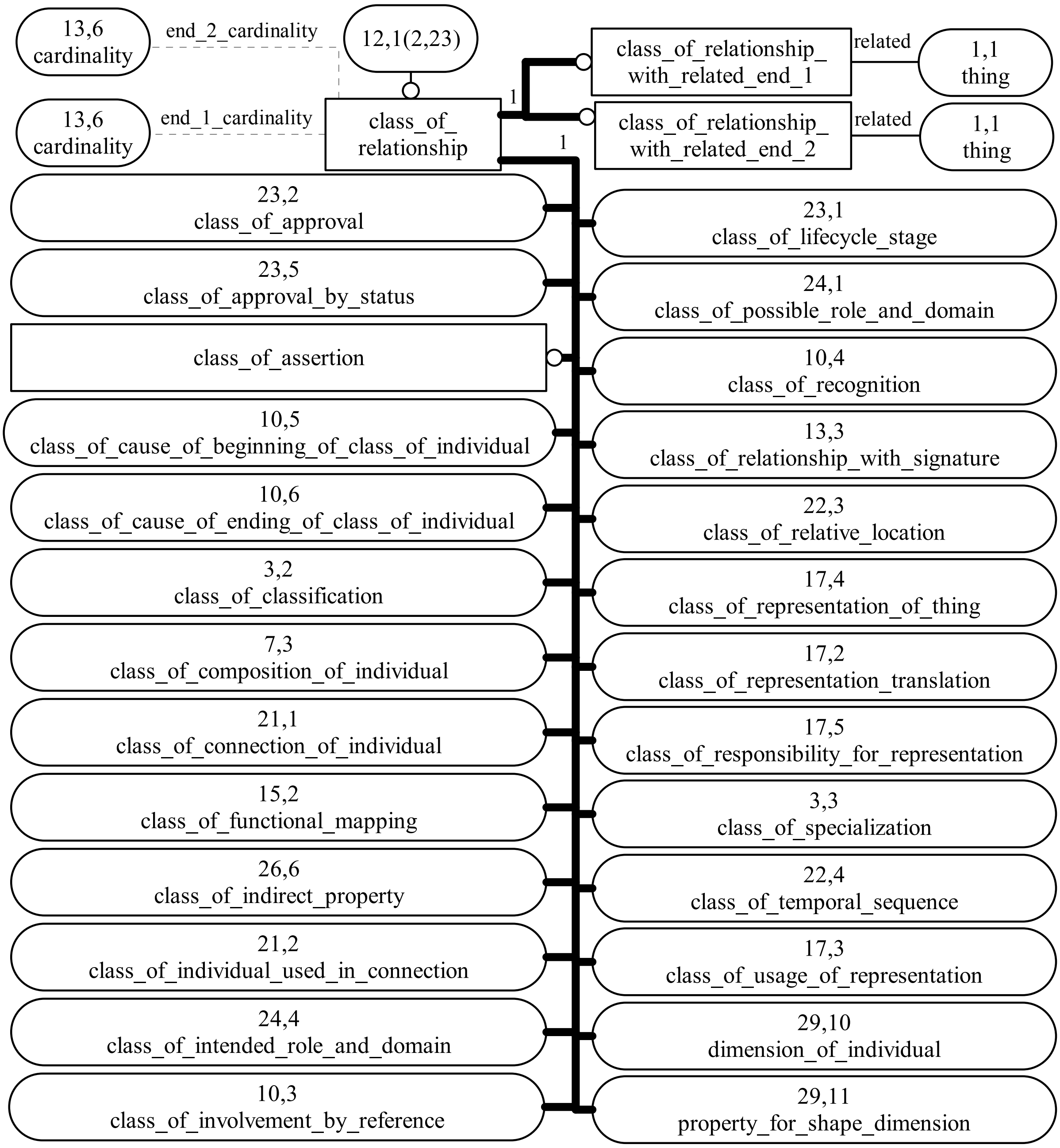

13,6
cardinality
end_2_cardinality
12,1(2,23)
class_of_relationship_
with_related_end_1
related
1,1
thing
13,6
cardinality
end_1_cardinality
class_of_
relationship
1
class_of_relationship_
with_related_end_2
related
1,1
thing
1
23,2
class_of_approval
23,1
class_of_lifecycle_stage
23,5
class_of_approval_by_status
24,1
class_of_possible_role_and_domain
class_of_assertion
10,4
class_of_recognition
10,5
class_of_cause_of_beginning_of_class_of_individual
13,3
class_of_relationship_with_signature
10,6
class_of_cause_of_ending_of_class_of_individual
22,3
class_of_relative_location
3,2
class_of_classification
17,4
class_of_representation_of_thing
7,3
class_of_composition_of_individual
17,2
class_of_representation_translation
21,1
class_of_connection_of_individual
17,5
class_of_responsibility_for_representation
15,2
class_of_functional_mapping
3,3
class_of_specialization
26,6
class_of_indirect_property
22,4
class_of_temporal_sequence
21,2
class_of_individual_used_in_connection
17,3
class_of_usage_of_representation
24,4
class_of_intended_role_and_domain
29,10
dimension_of_individual
10,3
class_of_involvement_by_reference
29,11
property_for_shape_dimension

Figure 188 — lifecycle_integration_schema EXPRESS-G diagram 12 of 29

5.2.12.1 class_of_assertion

A **class_of_assertion** is a **class_of_relationship** that describes the assertive nature of the member relations.

EXAMPLE Asserting, denying, and probabilistic can be represented by instances of **class_of_assertion**.

EXPRESS specification:

```
*)
  ENTITY class_of_assertion
    SUBTYPE OF(class_of_relationship);
  END_ENTITY;
(*
```

5.2.12.2 class_of_relationship

A **class_of_relationship** is a **class_of _abstract_object** whose members are members of **relationship**.

EXPRESS specification:

```
*)
  ENTITY class_of_relationship
  SUPERTYPE OF (ONEOF(
                              class_of_approval,
                              class_of_approval_by_status,
                              class_of_cause_of_beginning_of_class_of_individual,
                              class_of_cause_of_ending_of_class_of_individual,
                              class_of_classification,
                              class_of_composition_of_individual,
                              class_of_connection_of_individual,
                              class_of_functional_mapping,
                              class_of_indirect_property,
                              class_of_individual_used_in_connection,
                              class_of_intended_role_and_domain,
                              class_of_involvement_by_reference,
                              class_of_lifecycle_stage, class_of_assertion,
                              class_of_possible_role_and_domain,
                              class_of_recognition,
                              class_of_relationship_with_signature,
                              class_of_relative_location,
                              class_of_representation_of_thing,
                              class_of_representation_translation,
                              class_of_responsibility_for_representation,
                              class_of_specialization,
                              class_of_temporal_sequence,
                              class_of_usage_of_representation,
                              dimension_of_individual,
                              property_for_shape_dimension
                              )
                              ANDOR
                              ONEOF(class_of_relationship_with_related_end_1,
                              class_of_relationship_with_related_end_2))
    SUBTYPE OF(class_of_abstract_object);
    end_1_cardinality          :OPTIONAL cardinality;
    end_2_cardinality          :OPTIONAL cardinality;
  END_ENTITY;
 (*
```

Attribute definitions:

end_1_cardinality : The maximum and minimum cardinality for the first attribute of the **class_of_relationship**.
If no cardinality is specified, then there is no constraint on the cardinality.

end_2_cardinality : The maximum and minimum cardinality for the second attribute in the **class_of_relationship**.
If no cardinality is specified then there is no constraint on the cardinality.

5.2.12.3 class_of_relationship_with_related_end_1

A **class_of_relationship_with_related_end_1** is a **class_of_relationship** where a particular **thing** is related in the **class_of_relationship**, rather than the members of a **class**. The related **thing** plays the **role_and_domain** indicated by the **class_of_end_1**

EXAMPLE Products manufactured by Bloggs & Co is a **class_of_relationship** that points to Bloggs & Co as the related **thing**.

EXPRESS specification:

```
*)
  ENTITY class_of_relationship_with_related_end_1
    SUBTYPE OF(class_of_relationship);
    related                       :thing;
  END_ENTITY;
(*
```

Attribute definitions:

related : the particular **thing** that is related, and not some member of the **class** to which it may refer

5.2.12.4 class_of_relationship_with_related_end_2

A **class_of_relationship_with_related_end_2** is a **class_of_relationship** where a particular **thing** is related in the **class_of_relationship**, rather than the members of a **class**. The related **thing** plays the **role_and_domain** indicated by the **class_of_end_2**.

EXAMPLE Possession of welding skills by John Doe is an example of **class_of_relationship_with_related_end_2**, where John Doe is the related thing.

EXPRESS specification:

```
*)
  ENTITY class_of_relationship_with_related_end_2
    SUBTYPE OF(class_of_relationship);
    related                       :thing;
  END_ENTITY;
(*
```

Attribute definitions:

related : the particular **thing** that is related, and not some member of the **class** to which it may refer

5.2.13 Roles and domains

This subclause contains the declarations of entity data types that represent roles and domains.

NOTE Figure 189 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4.8).

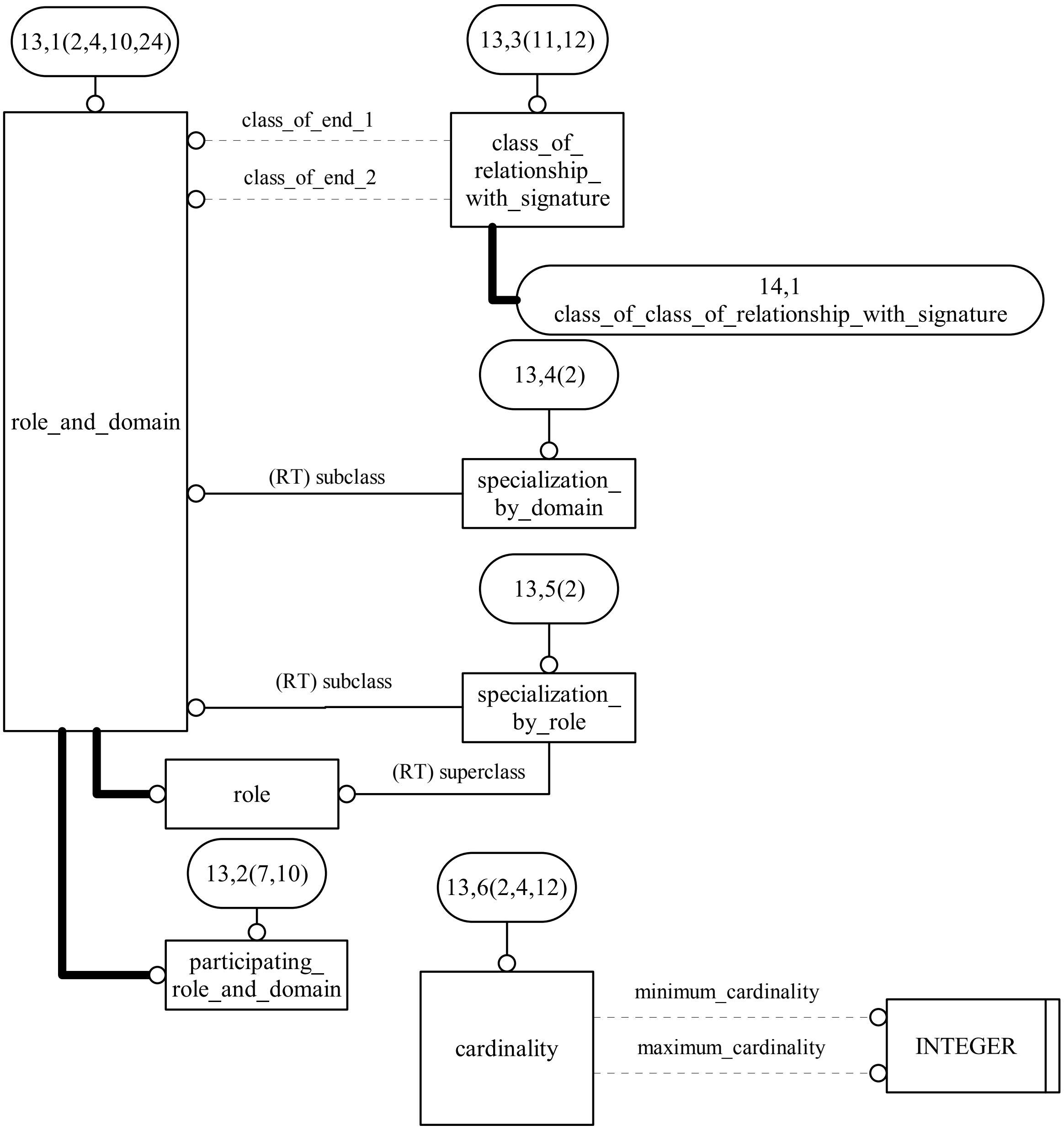


Figure 189 — lifecycle_integration_schema EXPRESS-G diagram 13 of 29

5.2.13.1 cardinality

A **cardinality** is a **class** that is the maximum and/or minimum number of times a thing can play a particular role in a **class_of_relationship** or **class_of_multidimensional_object**.

EXAMPLE A minimum of 1 and a maximum of 1 means that there is exactly one **relationship** or **multidimensional_object** of this type for each object.

EXPRESS specification:

```
*)
  ENTITY cardinality
    SUBTYPE OF(class);
    maximum_cardinality          :OPTIONAL INTEGER;
    minimum_cardinality          :OPTIONAL INTEGER;
  END_ENTITY;
(*
```

Attribute definitions:

maximum_cardinality : The maximum number of times a member of the domain can participate in the role specified.
If no **maximum_cardinality** is specified, then there is no maximum constraint.

NOTE 1 Common values for **maximum_cardinality** are 1 and many. Many is the result of specifying no value.

minimum_cardinality : The **minimum_cardinality** is the minimum number of times a member of the domain class may participate in the role specified.
If no **minimum_cardinality** is specified the value shall be taken as zero.

NOTE 2 Common values for the **minimum_cardinality** are zero and one.

5.2.13.2 class_of_relationship_with_signature

A **class_of_relationship_with_signature** is a **class_of_relationship** that may have a **role_and_domain** specified for each end.

NOTE A **class_of_relationship_with_signature** is analogous to a simple EXPRESS attribute and its inverse. More complex objects can be modelled with **multidimensional_object** and **class_of_multidimensional_object**.

EXAMPLE ‘Married’ is a **class_of_relationship** where **class_of_end_1** is the **role_and_domain** ‘husband’, and **class_of_end_2** the **role_and_domain** ‘wife’.

EXPRESS specification:

```
*)
  ENTITY class_of_relationship_with_signature
    SUBTYPE OF(class_of_relationship, relationship);
    class_of_end_1               :OPTIONAL role_and_domain;
    class_of_end_2               :OPTIONAL role_and_domain;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_end_1 : the specification of the **end_1** attribute of the members of the **class_of_relationship**

class_of_end_2 : the specification of the **end_2** attribute of the members of the **class_of_relationship**

5.2.13.3 participating_role_and_domain

A **participating_role_and_domain** is a **role_and_domain** that is also a **class_of_individual** that indicates a participating role in an **activity**.

EXAMPLE ‘Performer’ and ‘pumper’ are examples of **participating_role_and_domain**.

EXPRESS specification:

```
*)
  ENTITY participating_role_and_domain
    SUBTYPE OF(role_and_domain, class_of_individual);
  END_ENTITY;
(*
```

5.2.13.4 role

A **role** is a **role_and_domain** that indicates what some thing has to do with an **activity**, **relationship**, or **multidimensional_object**.

EXAMPLE 1 Employee is a **role** that indicates what a temporal part of a person has to do with an employment relation.

EXAMPLE 2 Pumper is a **role** that indicates what a temporal part of a pump has to do with a pumping activity.

EXPRESS specification:

```
*)
  ENTITY role
    SUBTYPE OF(role_and_domain);
  END_ENTITY;
(*
```

5.2.13.5 role_and_domain

A **role_and_domain** is a **class** that specifies the domainand and role for an end of a **class_of_relationship**, or **class_of_multidimensional_object**.

NOTE A **role_and_domain** is analogous to specifying an EXPRESS attribute or its inverse.

EXAMPLE "Husband and man" and "wife and woman" are examples of **role_and_domain**.

EXPRESS specification:

```
*)
  ENTITY role_and_domain
    SUBTYPE OF(class);
  END_ENTITY;
(*
```

5.2.13.6 specialization_by_domain

A **specialization_by_domain** is a **specialization** that indicates that the member of the **role_and_domain** is a **specialization** of the domain **class**.

EXAMPLE ‘Manufacturing company’ is a specialization of the ‘company’ domain.

EXPRESS specification:

```
*)
  ENTITY specialization_by_domain
    SUBTYPE OF(specialization);
    SELF\specialization.subclass: role_and_domain;
  END_ENTITY;
(*
```

Attribute definitions:

subclass : the **role_and_domain** that is the subclass of the **class**

5.2.13.7 specialization_by_role

A **specialization_by_role** is a **specialization** that indicates that the **role_and_domain** is of the **role** indicated by the superclass.

EXAMPLE Manufacturing company is a specialization by role of manufacturer.

EXPRESS specification:

```
*)
  ENTITY specialization_by_role
    SUBTYPE OF(specialization);
    SELF\specialization.subclass: role_and_domain;
    SELF\specialization.superclass      :    role;
  END_ENTITY;
(*
```

Attribute definitions:

subclass : the **role_and_domain** that is the subclass in the **specialization_by_role**

superclass : the **role** that is the superclass in the **specialization_by_role**

5.2.14 Classes of class of relationship

This subclause contains the declarations of entity data types that represent classes of class of relationship.

NOTE Figure 190 is a diagram of the entity data type(s) defined in this subclause.

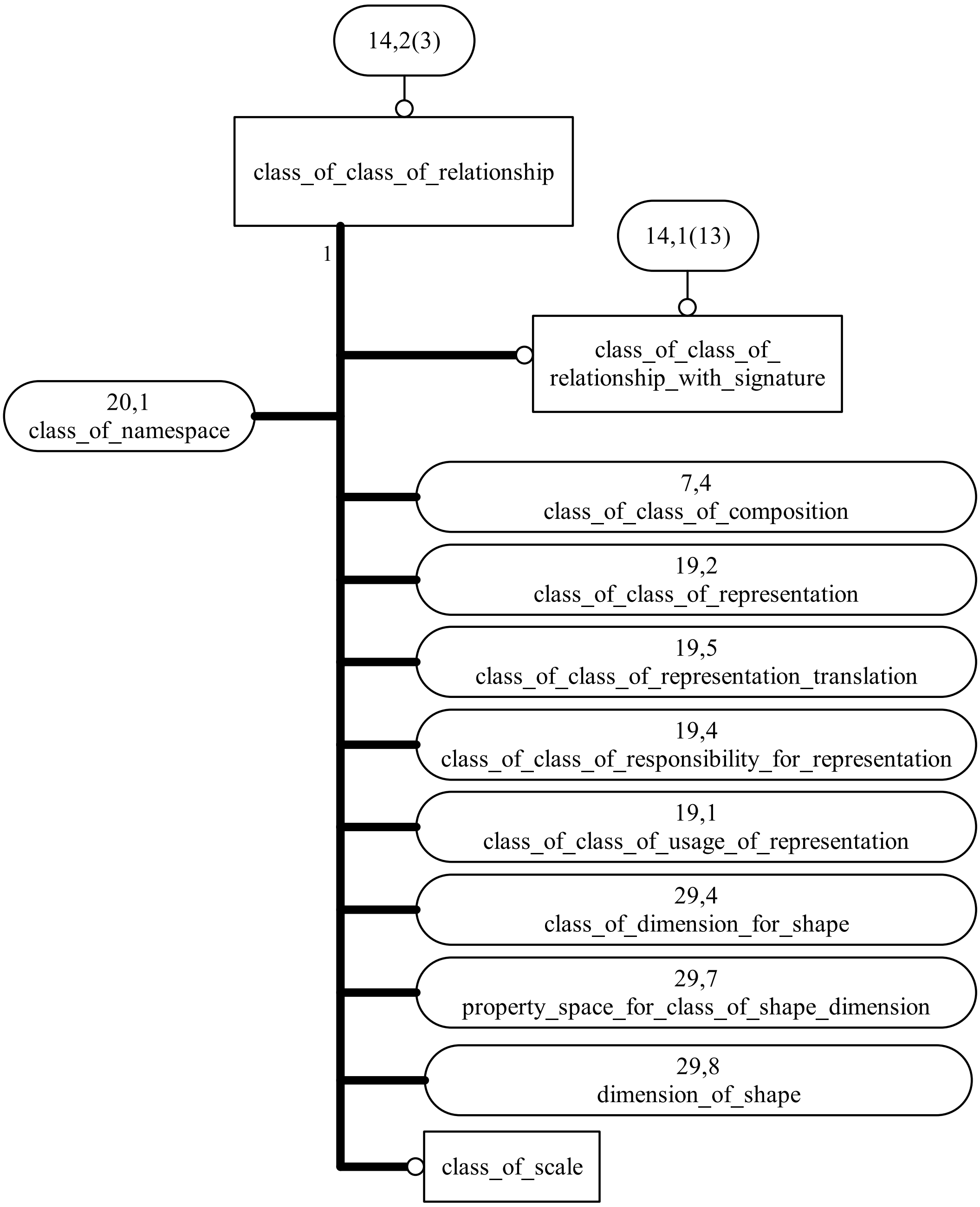

14,2(3)
class_of_class_of_relationship
1
14,1(13)
class_of_class_of_
relationship_with_signature
20,1
class_of_namespace
7,4
class_of_class_of_composition
19,2
class_of_class_of_representation
19,5
class_of_class_of_representation_translation
19,4
class_of_class_of_responsibility_for_representation
19,1
class_of_class_of_usage_of_representation
29,4
class_of_dimension_for_shape
29,7
property_space_for_class_of_shape_dimension
29,8
dimension_of_shape
class_of_scale

Figure 190 — lifecycle_integration_schema EXPRESS-G diagram 14 of 29

5.2.14.1 class_of_class_of_relationship

A **class_of_class_of_relationship** is a **class_of_class** whose members are instances of **class_of_relationship**.

EXAMPLE ‘Reflexive’ is an example of **class_of_class_of_relationship**. A ‘Reflexive’ **class_of_relationship** is one that may have the same **thing** playing both roles, such as connection, where something may be connected to itself.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_relationship
 SUPERTYPE OF (ONEOF(
                  class_of_class_of_composition,
                  class_of_class_of_relationship_with_signature,
                  class_of_class_of_representation,
                  class_of_class_of_representation_translation,
                  class_of_class_of_responsibility_for_representation,
                  class_of_class_of_usage_of_representation,
                  class_of_dimension_for_shape,
                  class_of_namespace,
                  class_of_scale,
                  dimension_of_shape,
                  property_space_for_class_of_shape_dimension
                  ))
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.14.2 class_of_class_of_relationship_with_signature

An **class_of_class_of_relationship_with_signature** is a **class_of_class_of_relationship** and **class_of_relationship_with_signature**. The purpose of **class_of_class_of_relationship_with_signature** is to allow other types of classes of relationship, not explicitly defined as entity data types in this part of ISO 15926, to be defined as reference data.

EXAMPLE Transitive, with the roles from and to indicating the direction of transitivity, is a **class_of_class_of_relationship_with_signature**. A **class_of_relationship** is transitive if when A relates to B and B relates to C then A relates to C, all in the same way.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_relationship_with_signature
    SUBTYPE OF(class_of_class_of_relationship,
                      class_of_relationship_with_signature);
  END_ENTITY;
(*
```

5.2.14.3 class_of_scale

A **class_of_scale** is a **class_of_class_of_relationship** whose members are instances of **scale**.

EXAMPLE SI Unit is an example of **class_of_scale**.

EXPRESS specification:

```
*)
  ENTITY class_of_scale
    SUBTYPE OF(class_of_class_of_relationship);
  END_ENTITY;
(*
```

5.2.15 Functions

This subclause contains the declarations of entity data types that represent functions.

NOTE Figure 191 is a diagram of the entity data type(s) defined in this subclause (see also 4.9).

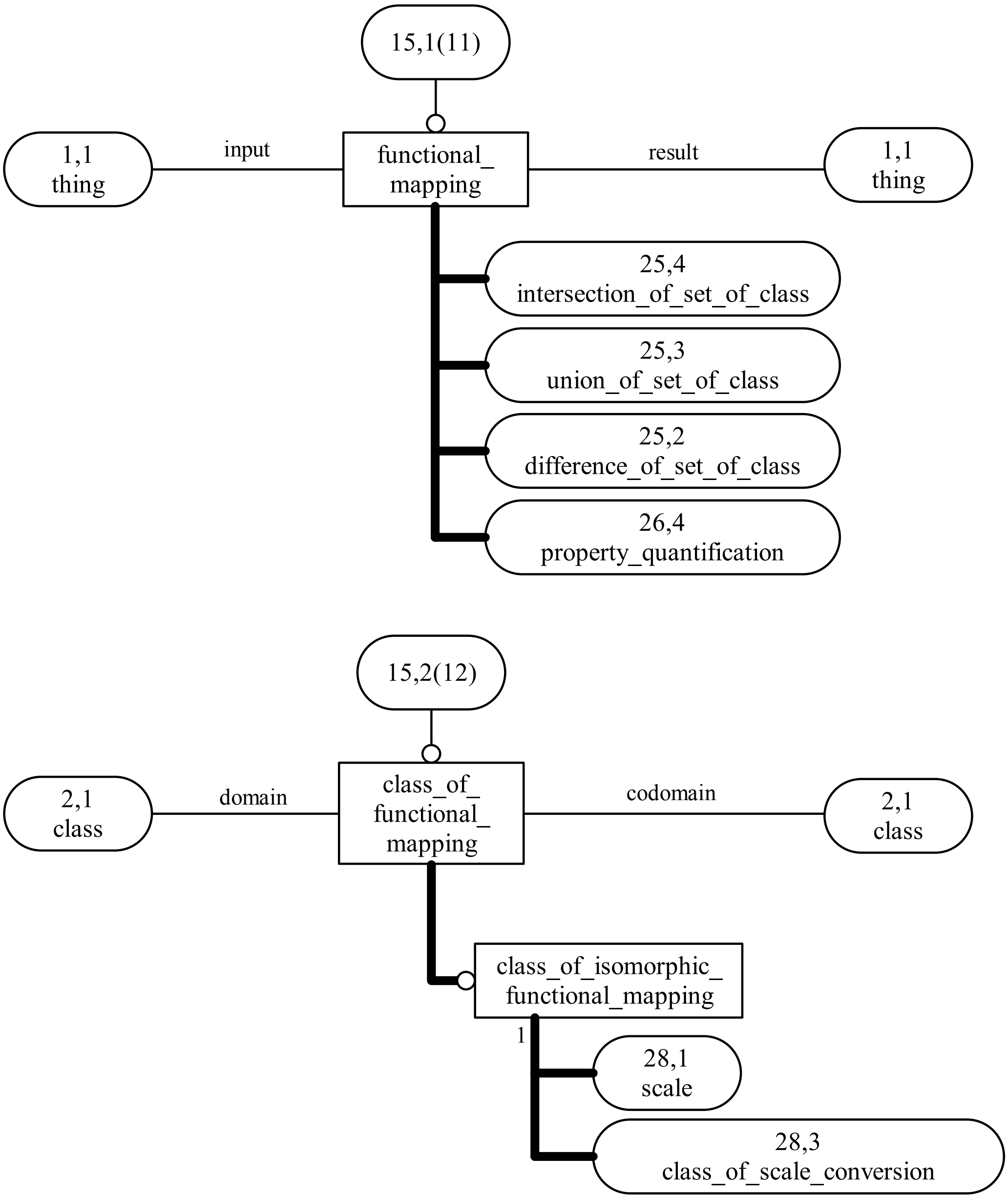

15,1(11)
1,1 thing
input
functional_ mapping
result
1,1 thing
25,4 intersection_of_set_of_class
25,3 union_of_set_of_class
25,2 difference_of_set_of_class
26,4 property_quantification
15,2(12)
2,1 class
domain
class_of_ functional_ mapping
codomain
2,1 class
class_of_isomorphic_ functional_mapping
1
28,1 scale
28,3 class_of_scale_conversion

Figure 191 — lifecycle_integration_schema EXPRESS-G diagram 15 of 29

5.2.15.1 class_of_functional_mapping

A **class_of_functional_mapping** is a **class_of_relationship** that is a many to one mapping. A **class_of_functional_mapping** is a function.

NOTE 1 This entity type would naturally have the name of function, but this is an EXPRESS reserved word.

NOTE 2 The significance of a function being a many to one mapping is that the same answer is always obtained. So, for example, 5 - 3 always gives 2. The minus function on two other arguments can also give 2.

NOTE 3 When there are several arguments to a function, then these are presented in a **multidimensional_object**.

EXAMPLE Minus is an example of **class_of_functional_mapping**.

EXPRESS specification:

```
*)
  ENTITY class_of_functional_mapping
    SUBTYPE OF(class_of_relationship);
    codomain                   :class;
    domain                     :class;
  END_ENTITY;
(*
```

Attribute definitions:

codomain : the result of applying the function to the domain

domain : the set of things to which the function is applied

5.2.15.2 class_of_isomorphic_functional_mapping

A **class_of_isomorphic_functional_mapping** is a **class_of_functional_mapping** that is isomorphic.

EXAMPLE The natural logarithm function is a **class_of_isomorphic_functional_mapping**.

EXPRESS specification:

```
*)
  ENTITY class_of_isomorphic_functional_mapping
    SUPERTYPE OF (ONEOF(scale, class_of_scale_conversion))
    SUBTYPE OF(class_of_functional_mapping);
  END_ENTITY;
(*
```

5.2.15.3 functional_mapping

A **functional_mapping** is a **relationship** that indicates that the input gave the result as determined by the classifying **class_of_functional_mapping**.

EXAMPLE The mapping of [5, 3] to 2, classified by the minus function is an example of **functional_mapping**.

EXPRESS specification:

```
*)
  ENTITY functional_mapping
    SUBTYPE OF(relationship);
    input                      :thing;
    result                     :thing;
  END_ENTITY;
(*
```

Attribute definitions:

input : the input to the mapping

result : the result of the application of the function to the input

5.2.16 Representations of things

This subclause contains the declarations of entity data types that represent representations of things.

NOTE Figure 192 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4.2).

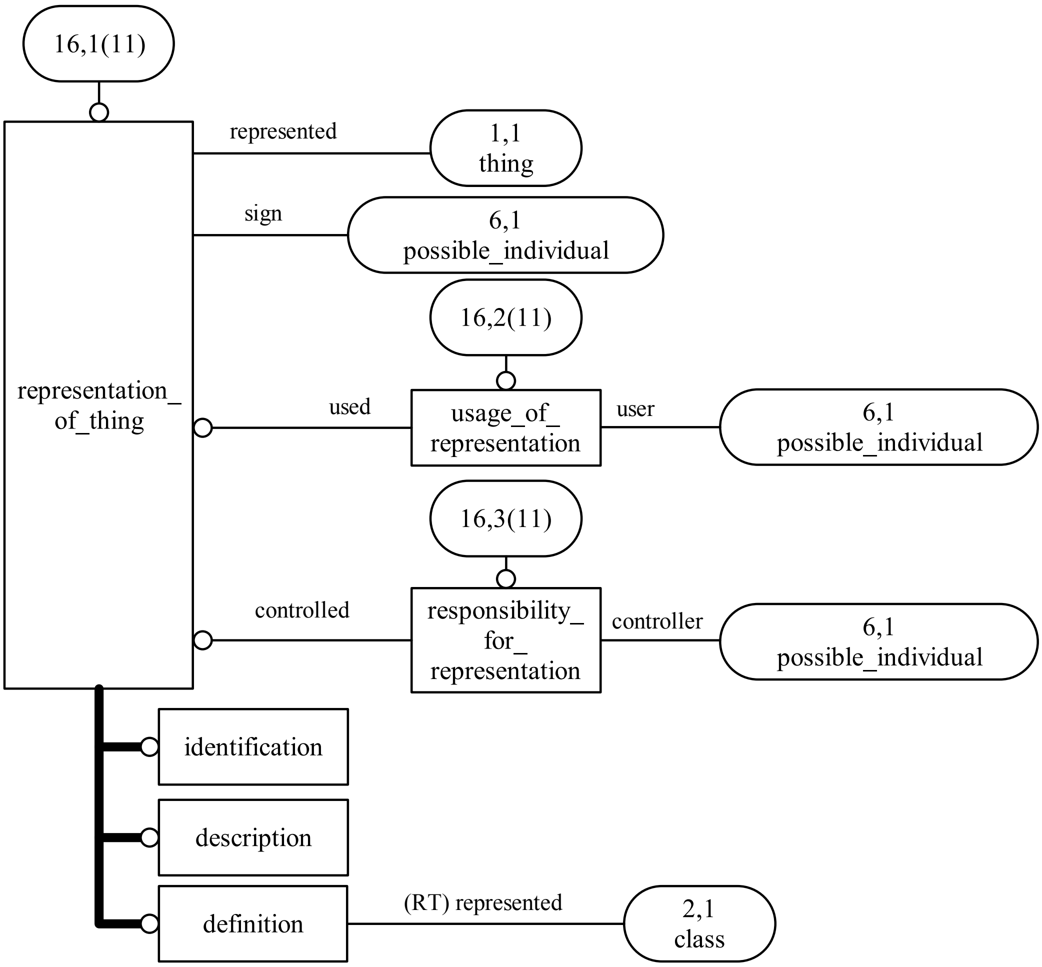


Figure 192 — lifecycle_integration_schema EXPRESS-G diagram 16 of 29

5.2.16.1 definition

A **definition** is a **representation_of_thing** that indicates that the **class** is defined by the sign **possible_individual**.

EXAMPLE The **relationship** between this copy of the preceding sentence and the heading before that is a **definition**.

EXPRESS specification:

```
*)
  ENTITY definition
    SUBTYPE OF(representation_of_thing);
    SELF\representation_of_thing.represented :     class;
  END_ENTITY;
(*
```

Attribute definitions:

represented : the **class** that is defined in the **definition**

5.2.16.2 description

A **description** is a **representation_of_thing** that indicates that the **possible_individual** describes the **thing**.

EXAMPLE A copy of the Piping and Instrumentation Diagram for Crude Distillation Unit 1 at refinery X has a **description** relationship with the plant.

EXPRESS specification:

```
*)
  ENTITY description
    SUBTYPE OF(representation_of_thing);
  END_ENTITY;
(*
```

5.2.16.3 identification

An **identification** is a **representation_of_thing** that indicates that the **possible_individual** is an identifier for the **thing** identified.

EXAMPLE 1 The relationship between the text "P101" on a printed copy of a pump data sheet and the applicable **functional_physical_object** is an example of **identification**.

EXAMPLE 2 The relationship between a name tag and an employee wearing it is an example of **identification**.

EXPRESS specification:

```
*)
  ENTITY identification
    SUBTYPE OF(representation_of_thing);
  END_ENTITY;
(*
```

5.2.16.4 representation_of_thing

A **representation_of_thing** is a **relationship** that indicates that a **possible_individual** is a sign for a **thing**.

EXAMPLE The relationship between a nameplate with its serial number and other data, and a particular pressure vessel (**materialized_physical_object**) is an example of **representation_of_thing** that is an **identification**.

NOTE In general it will be **class_of_representation_of_thing** that will be of interest, rather than each **representation_of_thing**. However, **representation_of_thing** will be of interest when individual copies of documents are managed and controlled.

EXPRESS specification:

```
*)
  ENTITY representation_of_thing
    SUBTYPE OF(relationship);
    represented                 :thing;
    sign                        :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

represented : the **thing** that is represented in the **representation_of_thing**

sign : the **possible_individual** that is the sign in the **representation_of_thing**

5.2.16.5 responsibility_for_representation

A **responsibility_for_representation** is a **relationship** that indicates that the controller **possible_individual** administers the controlled **representation_of_thing**.

EXAMPLE The responsibility for the administration of this part of ISO 15926 lies with ISO.

EXPRESS specification:

```
*)
  ENTITY responsibility_for_representation
    SUBTYPE OF(relationship);
    controlled                  :representation_of_thing;
    controller                  :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

controlled : the **representation_of_thing** controlled in the **responsibility_for_representation**

controller : the **possible_individual** that is the controller in the **responsibility_for_representation**

5.2.16.6 usage_of_representation

A **usage_of_representation** is a **relationship** that indicates that the **representation_of_thing** is used by the **possible_individual**. Usage does not imply responsibility.

EXAMPLE The sign "P101" is used by the XYZ company to represent a particular pump in a design.

EXPRESS specification:

```
*)
  ENTITY usage_of_representation
    SUBTYPE OF(relationship);
    used                        :representation_of_thing;
    user                        :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

used : the **representation_of_thing** that is used by some user or user group

user : the **possible_individual** that is the user or user group that uses the **representation_of_thing**

5.2.17 Classes of representation

This subclause contains the declarations of entity data types that represent classes of representation.

NOTE Figure 193 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4.2).

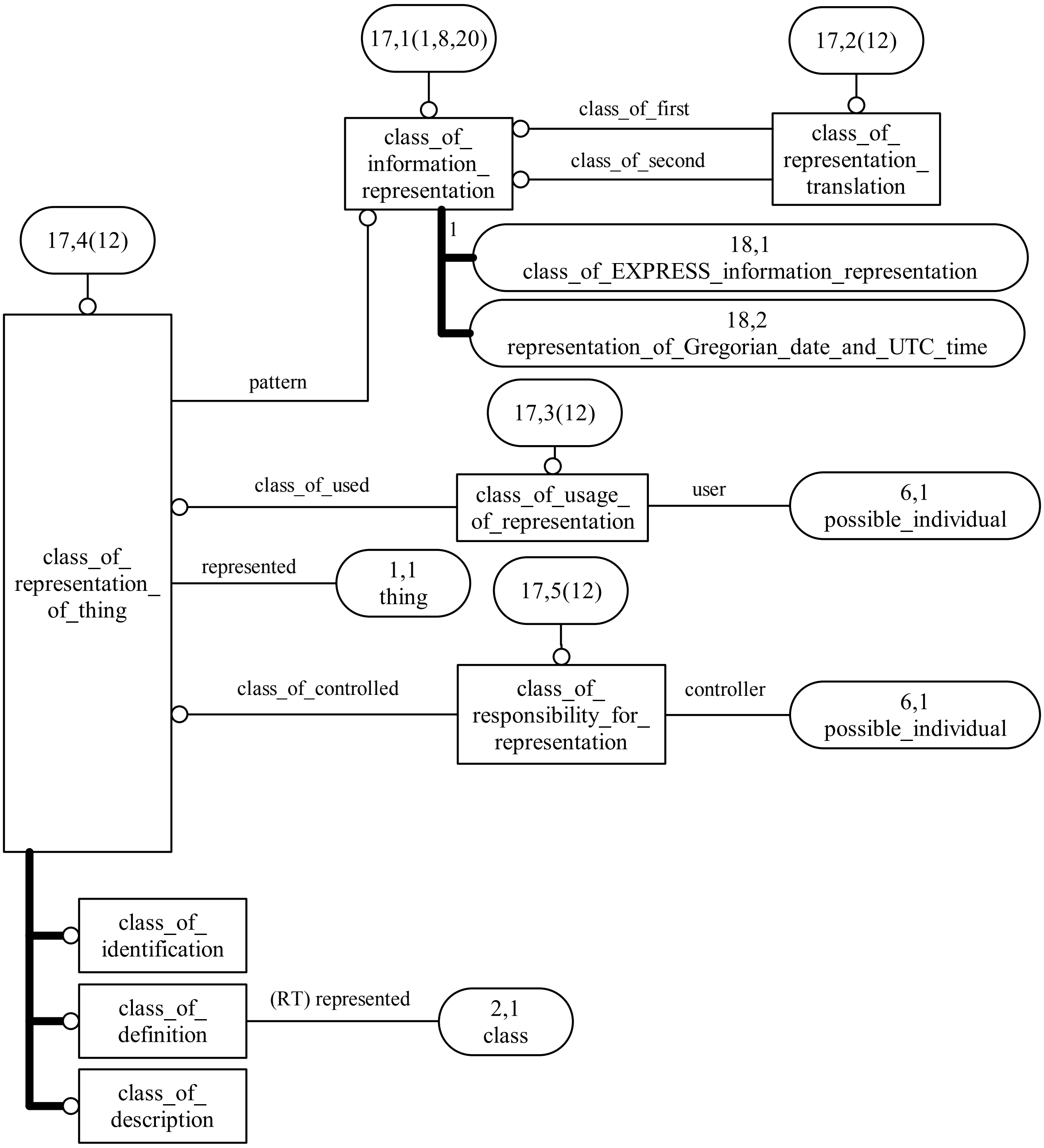


Figure 193 — lifecycle_integration_schema EXPRESS-G diagram 17 of 29

5.2.17.1 class_of_definition

A **class_of_definition** is a **class_of_representation_of_thing** that indicates the pattern is a definition of the represented **class**.

EXAMPLE The link between the pattern 'something that moves liquid' and the **class** that goes by the name ‘pump’ in English can be represented by an instance of **class_of_definition**.

EXPRESS specification:

```
*)
  ENTITY class_of_definition
```

```
    SUBTYPE OF(class_of_representation_of_thing);
    SELF\class_of_representation_of_thing.represented    :     class;
  END_ENTITY;
(*
```

Attribute definitions:

represented : the **class** that is defined by the members of the referenced **class_of_information_representation**

5.2.17.2 class_of_description

A **class_of_description** is a **class_of_representation_of_thing** that indicates the pattern is a description of the represented thing.

EXAMPLE The link between the pattern 'this is an old bilge pump' and a particular pump can be represented by an instance of **class_of_description**.

EXPRESS specification:

```
*)
  ENTITY class_of_description
    SUBTYPE OF(class_of_representation_of_thing);
  END_ENTITY;
(*
```

5.2.17.3 class_of_identification

A **class_of_identification** is a **class_of_representation_of_thing** that indicates that the pattern is used to refer to the represented thing.

EXAMPLE The link between the pattern 'AC-1234' and a particular pump, indicating that members of 'AC-1234' are used to refer to the pump, can be represented by an instance of **class_of_identification**.

EXPRESS specification:

```
*)
  ENTITY class_of_identification
    SUBTYPE OF(class_of_representation_of_thing);
  END_ENTITY;
(*
```

5.2.17.4 class_of_information_representation

A **class_of_information_representation** is a **class_of_arranged_individual** that defines a pattern that represents information.

EXAMPLE The texts formed with the pattern of characters 's' concatenated with 'u' concatenated with 'n' are members of the 'sun' **class_of_information_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_information_representation
    SUPERTYPE OF (ONEOF(class_of_EXPRESS_information_representation,
                  representation_of_Gregorian_date_and_UTC_time))
    SUBTYPE OF(class_of_arranged_individual);
  END_ENTITY;

(*
```

5.2.17.5 class_of_representation_of_thing

A **class_of_representation_of_thing** is a **class_of_relationship** that indicates that all members of the pattern **class_of_information_representation** represent the **thing**.

EXAMPLE The **class_of_relationship** that indicates that occurrences of the pattern denoted by 'London' represent the concept of the capital of the United Kingdom can be represented by an instance of **class_of_information_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_representation_of_thing
    SUBTYPE OF(class_of_relationship);
    pattern                    :class_of_information_representation;
    represented                :thing;
  END_ENTITY;
(*
```

Attribute definitions:

pattern : the **class_of_information_representation** whose members represent the referenced **thing**

represented : the **thing** that is represented by the members of the referenced **class_of_information_representation**

5.2.17.6 class_of_representation_translation

A **class_of_representation_translation** is a **class_of_relationship** that indicates the translation of two instances of **class_of_information_representation**.

EXAMPLE The link that indicates that the representations 'F' and '15' are equivalent (concept of fifteen in hexadecimal and octal respectively) can be represented by an instance of **class_of_representation_translation**.

EXPRESS specification:

```
*)
  ENTITY class_of_representation_translation
    SUBTYPE OF(class_of_relationship);
    class_of_first             :class_of_information_representation;
    class_of_second            :class_of_information_representation;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_first : the first instance of **class_of_information_representation** in the translation

class_of_second : the second instance of **class_of_information_representation** in the translation

5.2.17.7 class_of_responsibility_for_representation

A **class_of_responsibility_for_representation** is a **class_of_relationship** whose members indicate that a **possible_individual** (usually an organization) deems that members of the pattern can be used as representations of the represented thing.

EXAMPLE The link between the identification of pump #1234 and the XYZ Corporation, that indicates that the XYZ Corporation controls this identification, can be represented by an instance of **class_of_responsibility_for_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_responsibility_for_representation
    SUBTYPE OF(class_of_relationship);
    class_of_controlled        :class_of_representation_of_thing;
    controller                 :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_controlled : the **class_of_representation_of_thing** that is controlled by the referenced **possible_individual**

controller : the **possible_individual** that controls the referenced **class_of_representation_of_thing**

5.2.17.8 class_of_usage_of_representation

A **class_of_usage_of_representation** is a **class_of_relationship** whose members indicate that a **possible_individual** (usually an organization) reads or otherwise uses members of the pattern as a representation of the represented thing.

EXAMPLE The link between the identification of pump #1234 and contractor ABC Ltd, that indicates that ABC Ltd uses this identification can be represented by a class of **class_of_usage_of_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_usage_of_representation
    SUBTYPE OF(class_of_relationship);
    class_of_used              :class_of_representation_of_thing;
    user                       :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_used : the **class_of_representation_of_thing** that is used by the referenced **possible_individual**

user : the **possible_individual** that uses the referenced **class_of_representation_of_thing**

5.2.18 EXPRESS and UTC representations

This subclause contains the declarations of entity data types that represent EXPRESS and UTC representations.

NOTE Figure 194 is a diagram of the entity data type(s) defined in this subclause.

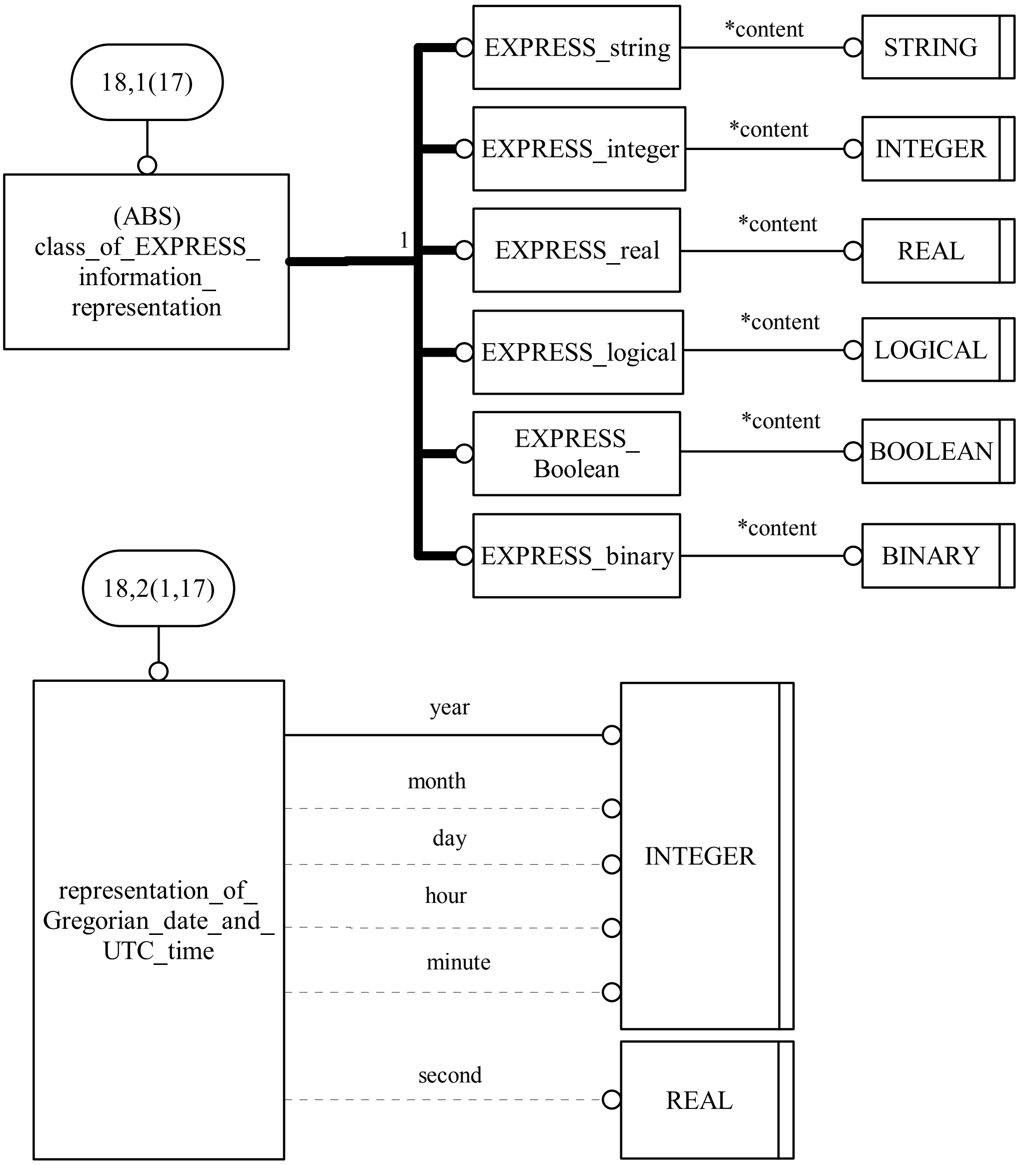


Figure 194 — lifecycle_integration_schema EXPRESS-G diagram 18 of 29

5.2.18.1 EXPRESS_Boolean

An **EXPRESS_Boolean** is a **class_of_EXPRESS_information_representation** that represents a Boolean value as defined in ISO 10303-11:1994, 8.1.5.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_Boolean
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                     :BOOLEAN;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_Boolean**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.2 EXPRESS_binary

An **EXPRESS_binary** is a **class_of_EXPRESS_information_representation** that represents a binary value as defined in ISO 10303-11:1994, 8.1.7.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_binary
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                     :BINARY;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_binary**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.3 EXPRESS_integer

An **EXPRESS_integer** is a **class_of_EXPRESS_information_representation** that represents an integer number as defined in ISO 10303-11:1994, 8.1.3.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_integer
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                    :INTEGER;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_integer**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.4 EXPRESS_logical

An **EXPRESS_logical** is a **class_of_EXPRESS_information_representation** that represents a logical value as defined in ISO 10303-11:1994, 8.1.4.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_logical
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                    :LOGICAL;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_logical**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.5 EXPRESS_real

An **EXPRESS_real** is a **class_of_EXPRESS_information_representation** that represents a real number as defined in ISO 10303-11:1994, 8.1.2.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_real
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                     :REAL;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_real**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.6 EXPRESS_string

An **EXPRESS_string** is a **class_of_EXPRESS_information_representation** that represents a string as defined in ISO 10303-11:1994, 8.1.6.

NOTE The UNIQUE rule ensures that any value is only held once.

EXPRESS specification:

```
*)
  ENTITY EXPRESS_string
    SUBTYPE OF(class_of_EXPRESS_information_representation);
    content                     :STRING;
  UNIQUE
    rule_1   :content;
  END_ENTITY;
(*
```

Attribute definitions:

content : the value of the **EXPRESS_string**

Formal proposition

rule_1 : the content shall be unique within a system

5.2.18.7 class_of_EXPRESS_information_representation

A **class_of_EXPRESS_information_representation** is a **class_of_information_representation** that is defined by ISO 10303-11.

EXPRESS specification:

```
*)
  ENTITY class_of_EXPRESS_information_representation
    ABSTRACT SUPERTYPE OF (ONEOF(EXPRESS_string, EXPRESS_integer,
                       EXPRESS_real, EXPRESS_logical, EXPRESS_Boolean,
                       EXPRESS_binary))
    SUBTYPE OF(class_of_information_representation);
  END_ENTITY;
(*
```

5.2.18.8 representation_of_Gregorian_date_and_UTC_time

A **representation_of_Gregorian_date_and_UTC_time** is a **class_of_information_representation** whose members are representations of time using the UTC system of time identification as specified in ISO 8601:2000 together with the Gregorian system for representing dates.

All times shall be represented using UTC representation of time. Dates shall follow the Gregorian calendar.

NOTE 1 Coordinated Universal Time (UTC) is the basis for legal time worldwide and follows TAI (see below) exactly except for an integral number of seconds, presently 32. These leap seconds are inserted on the advice of the International Earth Rotation Service (IERS) (http://hpiers.obspm.fr) to ensure that, on average over the years, the Sun is overhead within 0.9 seconds of 12:00:00 UTC on the meridian of Greenwich. UTC is thus the modern successor of Greenwich Mean Time, GMT, which was used when the unit of time was the mean solar day. International Atomic Time (TAI) is calculated by the BIPM from the readings of more than 200 atomic clocks located in metrology institutes and observatories in more than 30 countries around the world. TAI is made available every month in the BIPM Circular T (ftp://62.161.69.5/pub/tai/publication). It is estimated that TAI does not lose or gain with respect to an imaginary perfect clock by more than about one tenth of a microsecond (0.0000001 second) per year.

NOTE 2 Although ISO 8601 allows two representations for the midnight **hour**, 0000 and 2400, this part of ISO 15926 restricts the representation to 0000.

NOTE 3 A **second** value up to but not including 61.0 allows for leap seconds. The mean solar time is determined by the rotation of the earth. Leap seconds are added or subtracted as required, usually in the middle or at the end of a year, and ensure that the legal time does not differ from the non-uniform mean solar time by more than one second, in spite of the variations of the earth's rotation.

EXPRESS specification:

```
*)
  ENTITY representation_of_Gregorian_date_and_UTC_time
    SUBTYPE OF(class_of_information_representation);
    year                          :INTEGER;
    month                         :OPTIONAL INTEGER;
    day                           :OPTIONAL INTEGER;
    hour                          :OPTIONAL INTEGER;
    minute                        :OPTIONAL INTEGER;
    second                        :OPTIONAL REAL;
  WHERE
    valid_month                   : {1<= month <= 12};
    valid_day                     : {1<= day <= 31};
    valid_hour                    : {0<= hour <= 23};
    valid_minute                  : {0<= minute <= 59};
    valid_second                  : {0.0 <= second < 61.0};
  END_ENTITY;
(*
```

Attribute definitions:

year : the year as defined in the Gregorian calendar. The year shall be completely and explicitly specified using as many digits as necessary to unambiguously convey the century and year within the century. Truncated year numbers shall not be used.

month : the position of the specified month in a year as defined in ISO 8601:2000, 5.2.1.

day : the value of day as defined in ISO 8601:2000, 5.2.1

hour : the hour element of a specified time on a 24 hour clock. Midnight shall be represented by the value zero.

minute : the minute element of a specified time.

second : the second element of a specified time.

Formal propositions:

valid_month : month shall be between 1 and 12 inclusive

valid_day : day shall be between 1 and 31 inclusive

valid_hour : hour shall be between 0 and 23 inclusive

valid_minute : minute shall be between 0 and 59 inclusive

valid_second : second shall be from 0.0 and up to but not including 61.0

5.2.19 Classes of class of representation

This subclause contains the declarations of entity data types that represent classes of class of representation.

NOTE Figure 195 is a diagram of the entity data type(s) defined in this subclause.

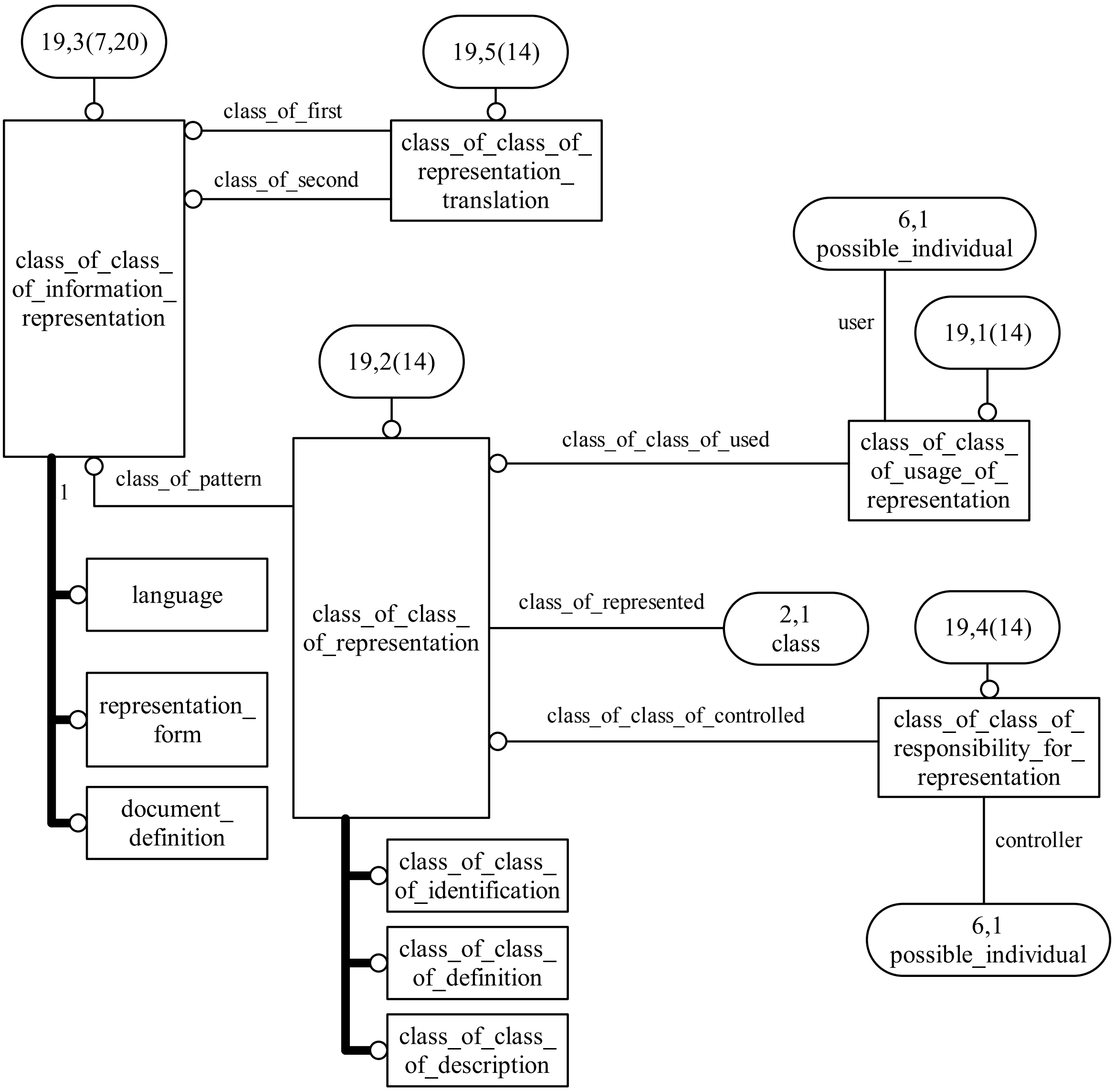

19,3(7,20)
19,5(14)
class_of_first
class_of_second
class_of_class_of_representation_translation
class_of_class_of_information_representation
6,1 possible_individual
user
19,1(14)
19,2(14)
class_of_class_of_used
class_of_class_of_usage_of_representation
class_of_pattern
1
language
representation_form
document_definition
class_of_class_of_representation
class_of_represented
2,1 class
19,4(14)
class_of_class_of_controlled
class_of_class_of_responsibility_for_representation
controller
6,1 possible_individual
class_of_class_of_identification
class_of_class_of_definition
class_of_class_of_description

Figure 195 — lifecycle_integration_schema EXPRESS-G diagram 19 of 29

5.2.19.1 class_of_class_of_definition

A **class_of_class_of_definition** is a **class_of_class_of_representation** whose members are members of **class_of_definition**.

EXAMPLE ‘Normative’ is a **class_of_class_of_definition**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_definition
    SUBTYPE OF(class_of_class_of_representation);
  END_ENTITY;
(*
```

5.2.19.2 class_of_class_of_description

A **class_of_class_of_description** is a **class_of_class_of_representation** whose members are members of **class_of_description**.

EXAMPLE Service description is a **class_of_class_of_description**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_description
    SUBTYPE OF(class_of_class_of_representation);
  END_ENTITY;
(*
```

5.2.19.3 class_of_class_of_identification

A **class_of_class_of_identification** is a **class_of_class_of_representation** whose members are members of **class_of_identification**.

EXAMPLE The link between the **class** 'family of manufactured parts' and the **representation_form** 'ISO 13584 Basic Semantic Unit', that indicates that part families can be identified using ISO 13584, can be represented by an instance of **class_of_class_of_identification**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_identification
    SUBTYPE OF(class_of_class_of_representation);
  END_ENTITY;
(*
```

5.2.19.4 class_of_class_of_information_representation

A **class_of_class_of_information_representation** is a **class_of_class_of_individual** that classifies information representation classes.

EXAMPLE Integer Octal is a **class_of_class_of_representation** whose members are all the information representation classes that correspond to Octal formatted integers.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_information_representation
    SUPERTYPE OF (ONEOF(representation_form, language,
                        document_definition))
    SUBTYPE OF(class_of_class_of_individual);
  END_ENTITY;
(*
```

5.2.19.5 class_of_class_of_representation

A **class_of_class_of_representation** is a **class_of_class_of_relationship** whose members are instances of **class_of_representation_of_thing**.

EXAMPLE The link that indicates that members of the class 'document' can be represented by patterns of the class 'XML' is a **class_of_class_of_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_pattern    :class_of_class_of_information_representation;
    class_of_represented       :class;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_pattern : the **class_of_class_of_information_representation** whose members can represent members of the referenced **class**

class_of_represented : the **class** whose members can be represented by members of the referenced **class_of_class_of_information_representation**

5.2.19.6 class_of_class_of_representation_translation

A **class_of_class_of_representation_translation** is a **class_of_class_of_relationship** whose members are members of **class_of_representation_translation**.

EXAMPLE The class ASCII whose members include all translation classes between members of the ASCII Binary and ASCII Text representation classes is a **class_of_class_of_representation_translation**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_representation_translation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_first     : class_of_class_of_information_representation;
    class_of_second    : class_of_class_of_information_representation;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_first : the first **class_of_class_of_information_representation** for which a translation is defined

class_of_second : the second **class_of_class_of_information_representation** for which a translation is defined

5.2.19.7 class_of_class_of_responsibility_for_representation

A **class_of_class_of_responsibility_for_representation** is a **class_of_class_of_relationship** whose members are members of **class_of_responsibility_for_representation** linking the controller to a set of representations.

EXAMPLE The link between Weir and the identification set between Weir pumps and Weir serial numbers indicating that the identifications are defined by Weir can be represented by an instance of **class_of_class_of_usage_of_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_responsibility_for_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_controlled: class_of_class_of_representation;
    controller                  :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_class_of_controlled : the **class_of_class_of_representation** that is controlled by the referenced **possible_individual**

controller : the **possible_individual** that controls the referenced **class_of_class_of_representation**

5.2.19.8 class_of_class_of_usage_of_representation

A **class_of_class_of_usage_of_representation** is a **class_of_class_of_relationship** whose members are members of **class_of_usage_of_representation** linking the user to a set of representations.

EXAMPLE The link between a user company and the identification set between Weir pumps and Weir serial numbers indicating that the user company uses the Weir identifiers can be represented by an instance of **class_of_class_of_usage_of_representation**.

EXPRESS specification:

```
*)
  ENTITY class_of_class_of_usage_of_representation
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_used     :class_of_class_of_representation;
    user                       :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_class_of_used : the **class_of_class_of_representation** that is used by the referenced **possible_individual**

user : the **possible_individual** that uses the referenced **class_of_class_of_representation**

5.2.19.9 document_definition

A **document_definition** is a **class_of_class_of_information_representation** that defines the content and/or structure of documents.

EXAMPLE XYZ Corp. Material Safety Data Sheet is a **document_definition**.

EXPRESS specification:

```
*)
  ENTITY document_definition
```

```
    SUBTYPE OF(class_of_class_of_information_representation);
  END_ENTITY;
(*
```

5.2.19.10 language

A **language** is a **class_of_class_of_information_representation** whose members are all the information representations made in the language.

EXAMPLE English, French, C++ and Java can be represented by instances of **language**.

EXPRESS specification:

```
*)
  ENTITY language
    SUBTYPE OF(class_of_class_of_information_representation);
  END_ENTITY;
(*
```

5.2.19.11 representation_form

A **representation_form** is a **class_of_class_of_information_representation** that distinguishes the form of representation.

EXAMPLE Hexadecimal, text, script, symbol, picture, diagram, semaphore, Morse code, music score, MIDI file format, and XML can each be represented by instances of **representation_form**.

EXPRESS specification:

```
*)
  ENTITY representation_form
    SUBTYPE OF(class_of_class_of_information_representation);
  END_ENTITY;
(*
```

5.2.20 Namespaces

This subclause contains the declarations of entity data types that represent namespaces.

NOTE Figure 196 is a diagram of the entity data type(s) defined in this subclause (see 4.8.4.2.5).

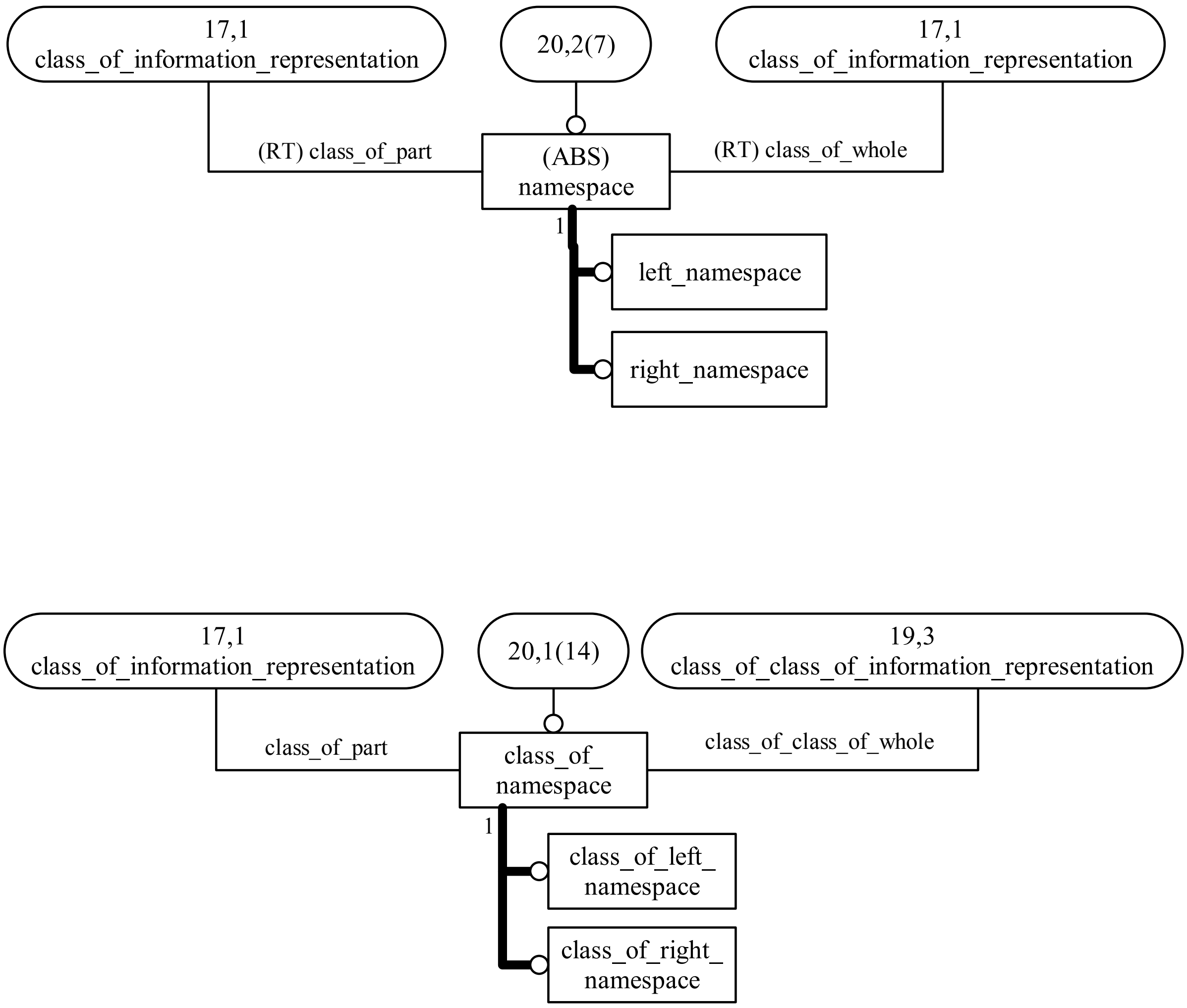


Figure 196 — lifecycle_integration_schema EXPRESS-G diagram 20 of 29

5.2.20.1 class_of_left_namespace

A **class_of_left_namespace** is a **class_of_namespace** that indicates that the class_of_part is the **left_namespace** for the members of the class_of_class_of_whole.

EXAMPLE WC1: is the **left_namespace** for customer site identifiers for Water Company 1.

EXPRESS specification:

```
*)
  ENTITY class_of_left_namespace
    SUBTYPE OF(class_of_namespace);
  END_ENTITY;
(*
```

5.2.20.2 class_of_namespace

A **class_of_namespace** is a **class_of_class_of_relationship** that indicates that a **class_of_information_representation** is the **class_of_part** used as a namespace for each member of a **class_of_class_of_information_representation** that is the **class_of_class_of_whole**.

EXAMPLE WC1: is used as the name space for a set of water company identifiers.

EXPRESS specification:

```
*)
  ENTITY class_of_namespace
    SUPERTYPE OF (ONEOF(class_of_left_namespace,
                      class_of_right_namespace))
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_class_of_whole    :
                        class_of_class_of_information_representation;
    class_of_part              :class_of_information_representation;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_class_of_whole : the **class_of_class_of_information_representation** whose members have the namespace

class_of_part : the **class_of_information_representation** that is the namespace

5.2.20.3 class_of_right_namespace

A **class_of_right_namespace** is a **class_of_namespace** where the **class_of_part** is the namespace for the members of the **class_of_class_of_whole**.

EXPRESS specification:

```
*)
  ENTITY class_of_right_namespace
    SUBTYPE OF(class_of_namespace);
  END_ENTITY;
(*
```

5.2.20.4 left_namespace

A **left_namespace** is a **namespace** where the **class_of_part** is the left part of the **class_of_whole**.

EXAMPLE Where WC1: is the **namespace** in WC1:1234, it is a **left_namespace**.

EXPRESS specification:

```
*)
  ENTITY left_namespace
    SUBTYPE OF(namespace);
  END_ENTITY;
(*
```

5.2.20.5 namespace

A **namespace** is a **class_of_arrangement_of_individual** where the **class_of_whole** and **class_of_part** are members of **class_of_information_representation** and the part is the most significant part of the whole that is the namespace.

EXAMPLE The STRING WC1: is the namespace in the identifier WC1:1234.

EXPRESS specification:

```
*)
  ENTITY namespace
    ABSTRACT SUPERTYPE OF (ONEOF(right_namespace, left_namespace))
    SUBTYPE OF(class_of_arrangement_of_individual);
    SELF\class_of_composition_of_individual.class_of_part      :
                                class_of_information_representation;
    SELF\class_of_arrangement_of_individual.class_of_whole     :
                                class_of_information_representation;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_part : the **class_of_information_representation** that is the namespace

class_of_whole : the **class_of_information_representation** that has the **class_of_part** as a namespace

5.2.20.6 right_namespace

A **right_namespace** is a **namespace** that indicates that the **class_of_part** is the right most part of the **class_of_whole**.

EXAMPLE When ZH is the namespace in 5367ZH, this is indicated by a **right_namespace** relationship between them.

EXPRESS specification:

```
*)
  ENTITY right_namespace
    SUBTYPE OF(namespace);
  END_ENTITY;
(*
```

5.2.21 Connections

This subclause contains the declarations of entity data types that represent connections.

NOTE Figure 197 is a diagram of the entity data type(s) defined in this subclause (see 4.7.3).

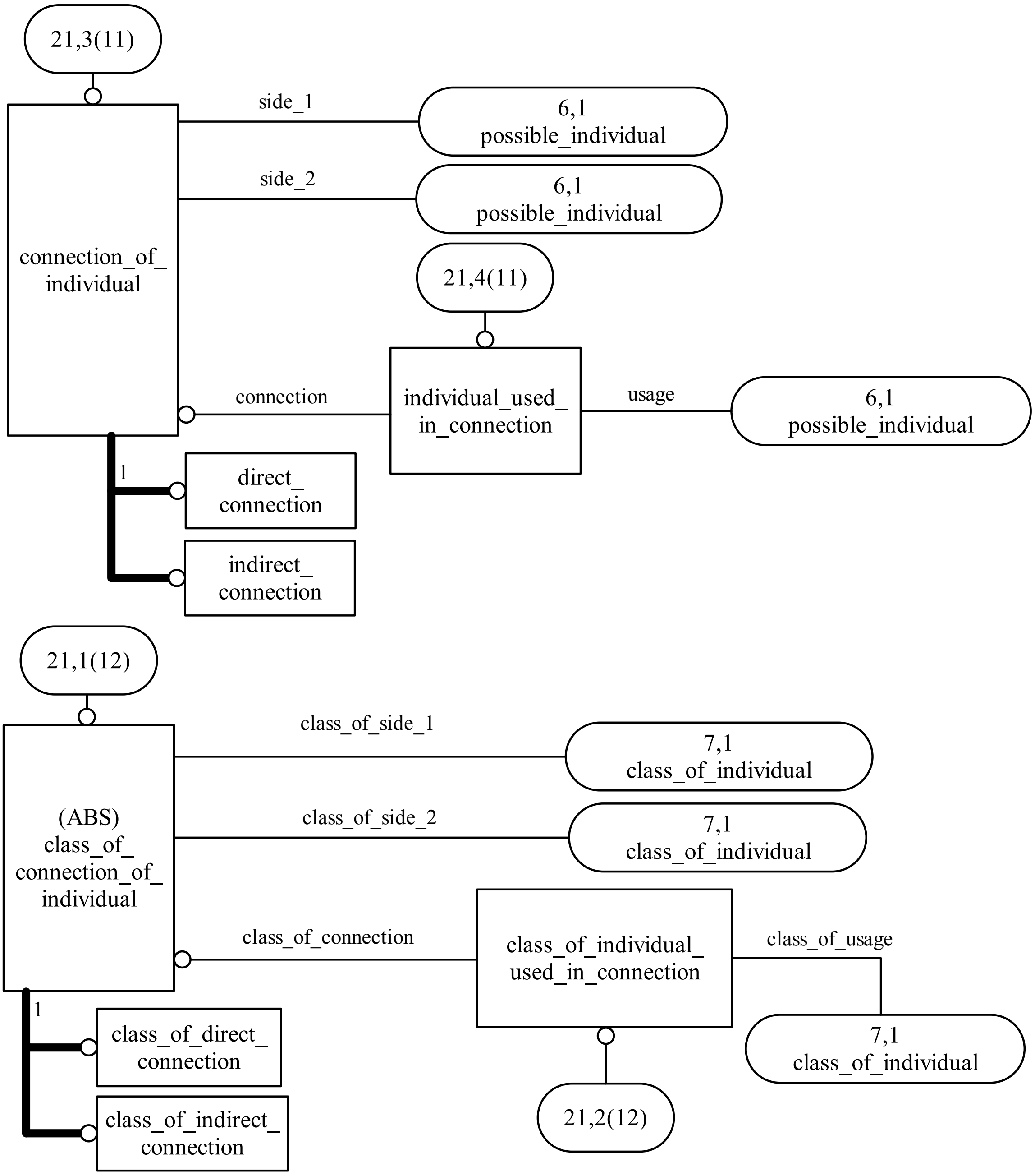


Figure 197 — lifecycle_integration_schema EXPRESS-G diagram 21 of 29

5.2.21.1 class_of_connection_of_individual

A **class_of_connection_of_individual** is a **class_of_relationship** whose members are members of **connection_of_individual**. It indicates that a member of the **class_of_side_1 class_of_individual** can be connected to a member of the **class_of_side_2 class_of_individual**.

NOTE 1 The **class_of_side_1** and **class_of_side_2** indicate the **class_of_individual** that is the **side_1** and **side_2** respectively in a **connection_of_individual** that is a member of this **class_of_connection_of_individual**.

NOTE 2 Flexible, rigid, and welded cannot be represented as instances of **class_of_connection_of_individual**, these are classes of the materials connected or used in the connection.

EXAMPLE Electrical connection between wires is a **class_of_connection_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_connection_of_individual
    ABSTRACT SUPERTYPE OF (ONEOF(class_of_direct_connection,
                    class_of_indirect_connection))
    SUBTYPE OF(class_of_relationship);
    class_of_side_1           :class_of_individual;
    class_of_side_2           :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_side_1 : the **class_of_individual** whose members play the role of **side_1** in the members of the **class_of_connection_of_individual**

class_of_side_2 : the **class_of_individual** whose members play the role of **side_2** in the members of the **class_of_connection_of_individual**

5.2.21.2 class_of_direct_connection

A **class_of_direct_connection** is a **class_of_connection_of_individual** whose members are members of **direct_connection**.

EXAMPLE Three-pin electrical plug into three-pin socket is an example of **class_of_direct_connection**.

EXPRESS specification:

```
*)
  ENTITY class_of_direct_connection
    SUBTYPE OF(class_of_connection_of_individual);
  END_ENTITY;
(*
```

5.2.21.3 class_of_indirect_connection

A **class_of_indirect_connection** is a **class_of_connection_of_individual** whose members are members of **indirect_connection**.

EXAMPLE Drip pipe indirectly connected to drain funnel is an example of **class_of_indirect_connection**.

EXPRESS specification:

```
*)
  ENTITY class_of_indirect_connection
    SUBTYPE OF(class_of_connection_of_individual);
  END_ENTITY;
(*
```

5.2.21.4 class_of_individual_used_in_connection

A **class_of_individual_used_in_connection** is a **class_of_relationship** whose members are members of **individual_used_in_connection**. It indicates that a member of the **class_of_individual** is used in a **class_of_connection_of_individual**.

EXAMPLE The link between the **class_of_connection_of_individual** that indicates that B12 type beams are connected to pipe hangers, and the **class_of_individual** "20mm Diameter bolt", that indicates that four 20mm diameter bolts are used in the connection of a pipe hanger to a type B12 beam can be represented by an instance of **class_of_individual_involved_in_connection**.

<u>EXPRESS specification:</u>

```
*)
  ENTITY class_of_individual_used_in_connection
    SUBTYPE OF(class_of_relationship);
    class_of_connection         :class_of_connection_of_individual;
    class_of_usage              :class_of_individual;
  END_ENTITY;
(*
```

<u>Attribute definitions:</u>

class_of_connection : the **class_of_connection_of_individual** whose members are the connections in the members of the **class_of_individual_involved_in_connection**

class_of_usage : the **class_of_individual** whose members are used in the members of the **class_of_individual_used_in_connection**

5.2.21.5 connection_of_individual

A **connection_of_individual** is a **relationship** that indicates that matter, energy, or both can be transferred between the members of **possible_individual** that are connected, either directly or indirectly. There is no significance to the ordering of the two related instances of **possible_individual**. The names **side_1** and **side_2** serve only to distinguish the attributes.

<u>EXPRESS specification:</u>

```
*)
  ENTITY connection_of_individual
    SUPERTYPE OF (ONEOF(direct_connection, indirect_connection))
    SUBTYPE OF(relationship);
    side_1                      :possible_individual;
    side_2                      :possible_individual;
  END_ENTITY;
(*
```

<u>Attribute definitions:</u>

side_1 : the first **possible_individual** that is involved in the **connection_of_individual**

side_2 : the second **possible_individual** that is involved in the **connection_of_individual**

5.2.21.6 direct_connection

A **direct_connection** is a **connection_of_individual** that indicates that the **side_1** and **side_2** are directly connected via a common spatial boundary.

EXAMPLE The relation that indicates that the plug terminating a serial communications cable is connected to the socket on a piece of computer equipment can be represented by an instance of of **direct_connection**.

EXPRESS specification:

```
*)
  ENTITY direct_connection
    SUBTYPE OF(connection_of_individual);
  END_ENTITY;
(*
```

5.2.21.7 indirect_connection

An **indirect_connection** is a **connection_of_individual** that indicates that side_1 and side_2 are connected via other individuals.

EXAMPLE The relation that indicates that there is a railway connection between the cities of London and Paris can be represented by an instance of **indirect_connection**.

EXPRESS specification:

```
*)
  ENTITY indirect_connection
    SUBTYPE OF(connection_of_individual);
  END_ENTITY;
(*
```

5.2.21.8 individual_used_in_connection

An **individual_used_in_connection** is a **relationship** that indicates that a **possible_individual** is used in a **connection_of_individual**.

EXAMPLE The **relationship** between the connection of the flanged ends of two pipes and a temporal part of the bolts, nuts, washers and gasket set that indicates that the bolt and gasket set participates in the connection can be represented by an instance of **individual_used_in_connection**.

EXPRESS specification:

```
*)
  ENTITY individual_used_in_connection
    SUBTYPE OF(relationship);
    connection                :connection_of_individual;
    usage                     :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

connection : the **connection_of_individual** in which the referenced **possible_individual** participates

usage : the **possible_individual** that participates in the referenced **connection_of_individual**

5.2.22 Relative locations and sequences

This subclause contains the declarations of entity data types that represent relative locations and sequences.

NOTE Figure 198 is a diagram of the entity data type(s) defined in this subclause (see also 4.7.4).

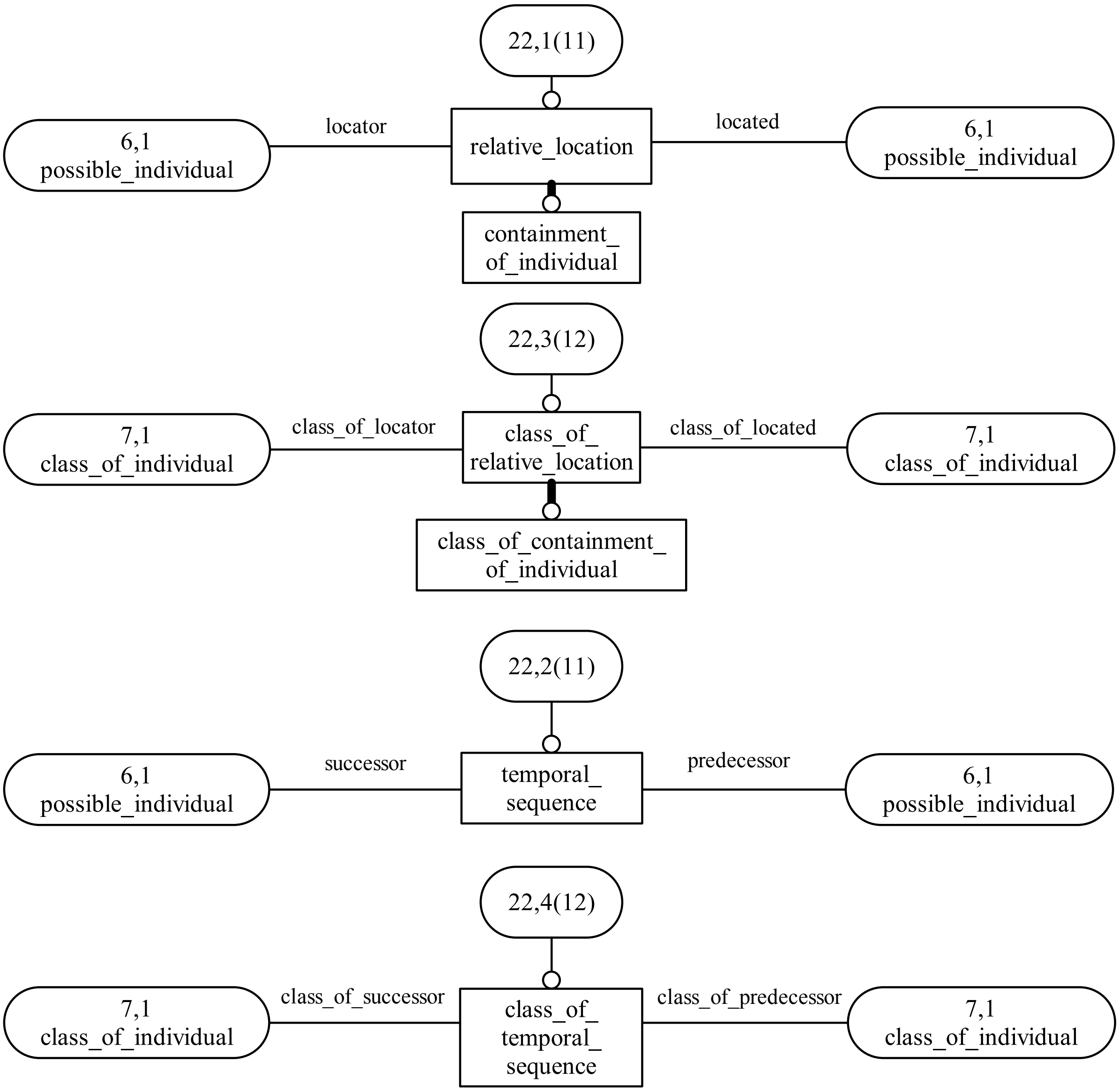


Figure 198 — lifecycle_integration_schema EXPRESS-G diagram 22 of 29

5.2.22.1 class_of_containment_of_individual

A **class_of_containment_of_individual** is a **class_of_relative_location** whose members are instances of **containment_of_individual**. It indicates that a member of the **class_of_locator class_of_individual** can contain a member of the **class_of_located class_of_individual**.

EXAMPLE That 'de-icing fluid' can be contained by a '1500ml screw-top plastic bottle' is a **class_of_containment_of_individual**.

EXPRESS specification:

```
*)
  ENTITY class_of_containment_of_individual
    SUBTYPE OF(class_of_relative_location);
  END_ENTITY;
(*
```

5.2.22.2 class_of_relative_location

A **class_of_relative_location** is a **class_of_relationship** whose members are instances of **relative_location**.

EXAMPLE Beside, above, and below are examples of **class_of_relative_location**.

EXPRESS specification:

```
*)
  ENTITY class_of_relative_location
    SUBTYPE OF(class_of_relationship);
    class_of_located           :class_of_individual;
    class_of_locator           :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_located : the **class_of_individual** whose members are located by members of the class_of_locator **class_of_individual**

class_of_locator : the **class_of_individual** whose members act as locator for the members of the class_of_located **class_of_individual**

5.2.22.3 class_of_temporal_sequence

A **class_of_temporal_sequence** is a **class_of_relationship** where the sequence is of a temporal nature.

EXAMPLE 1 The link that indicates that members of ‘July’ follow members of ‘June’ can be represented by an instance of **class_of_sequence**.

EXAMPLE 2 The link that indicates that emptying activities for a tank precede cleaning activities can be represented by an instance of **class_of_sequence**.

EXPRESS specification:

```
*)
  ENTITY class_of_temporal_sequence
    SUBTYPE OF(class_of_relationship);
    class_of_predecessor       :class_of_individual;
    class_of_successor         :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_predecessor : the **class_of_individual** whose members are the predecessors in the members of **class_of_sequence**

class_of_successor : the **class_of_individual** whose members are the successors in the members of **class_of_sequence**

5.2.22.4 containment_of_individual

A **containment_of_individual** is a **relative_location** where the located **possible_individual** is contained by the locator **possible_individual** but is not part of it.

EXAMPLE The contents of a vessel being inside the vessel can be represented by an instance of **containment_of_individual**.

NOTE Containment is distinct from composition; in composition the whole consists of all of its part, with containment, what is contained is not a part of the container.

EXPRESS specification:

```
*)
  ENTITY containment_of_individual
    SUBTYPE OF(relative_location);
  END_ENTITY;
(*
```

5.2.22.5 relative_location

A **relative_location** is a **relationship** that indicates that the position of one **possible_individual** is relative to another.

NOTE The **classification** of the **relative_location** indicates the nature of the **relative_location**, e.g. above, below, beside.

EXAMPLE A being the located relative to B being the locator in a **relative_location** that is classified by the **class_of_relative_location** above, indicates that A is above B.

EXPRESS specification:

```
*)
  ENTITY relative_location
    SUBTYPE OF(relationship);
    located                   :possible_individual;
    locator                   :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

located : the **possible_individual** that is located

locator : the **possible_individual** that is the reference location for the located **possible_individual**

5.2.22.6 temporal_sequence

A **temporal_sequence** is a **relationship** that indicates that one **possible_individual** precedes another in a temporal sense.

EXAMPLE 1 The **relationship** that indicates that the **possible_individual** that is the construction phase of a plant precedes the **possible_individual** that is the commissioning phase of a plant can be represented by an instance of **temporal_sequence**.

EXAMPLE 2 The **relationship** that indicates that the **period_in_time** known as the industrial revolution preceded the **period_in_time** known as the information revolution can be represented by an instance of **temporal_sequence**.

EXPRESS specification:

```
*)
  ENTITY temporal_sequence
    SUBTYPE OF(relationship);
    predecessor                 :possible_individual;
    successor                   :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

predecessor : the **possible_individual** that is the predecessor in the **sequence**

successor : the **possible_individual** that is the successor in a **sequence**

5.2.23 Lifecycle stages and approvals

This subclause contains the declarations of entity data types that represent lifecycle stages and approvals.

NOTE Figure 199 is a diagram of the entity data type(s) defined in this subclause (see 4.7.7 and 4.7.18).

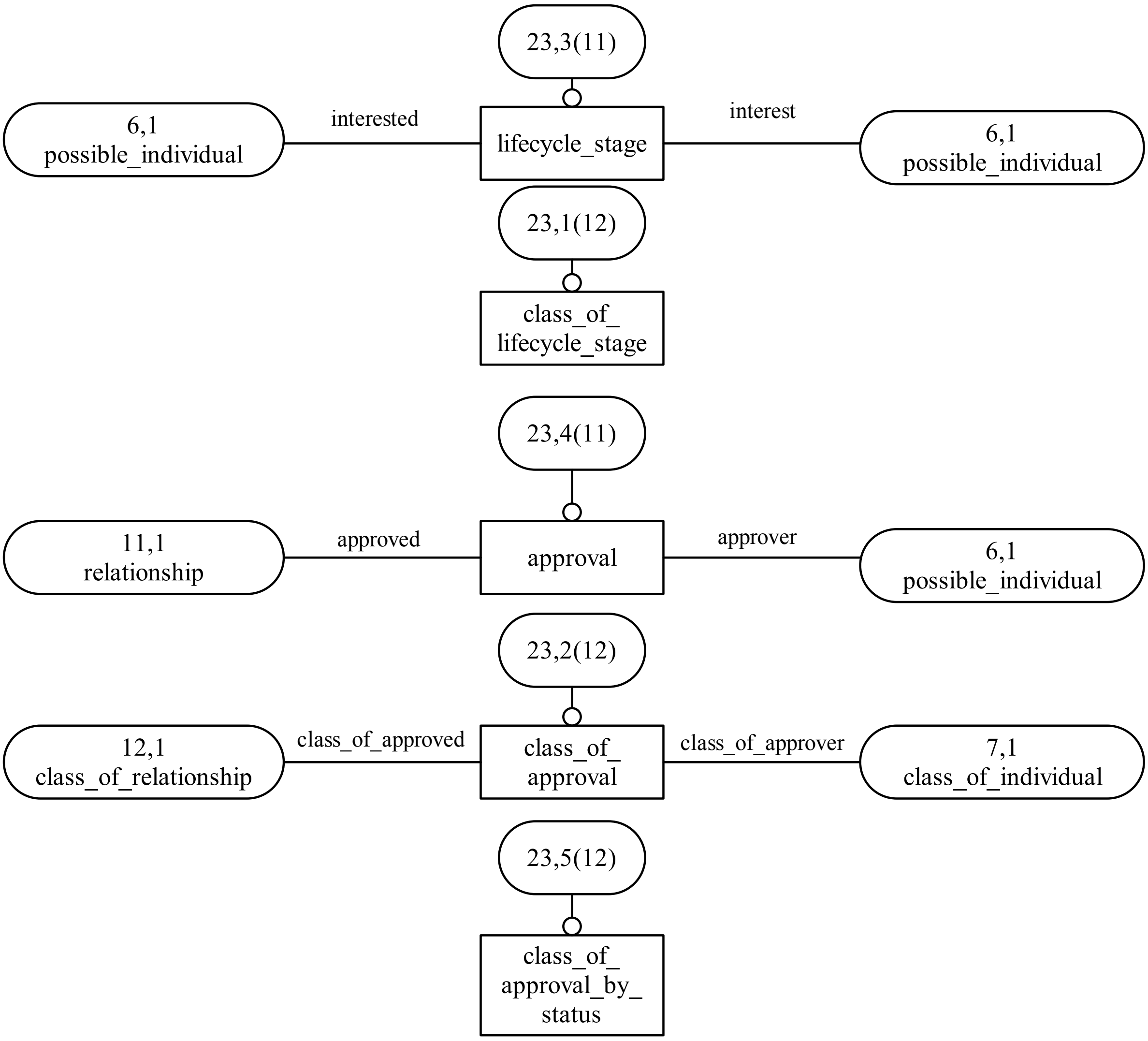


Figure 199 — lifecycle_integration_schema EXPRESS-G diagram 23 of 29

5.2.23.1 approval

An **approval** is a **relationship** that indicates that a **relationship** has been approved by a **possible_individual** that is an approver.

NOTE Care should be taken as to what is approved. Sometimes it will not be say a pump that is approved, but the participation of the pump in a particular **activity**, or member of some **class_of_activity**.

EXAMPLE The **involvement_by_reference** of a plant design with a construction activity, being approved by the site manager, is an example of an **approval**.

EXPRESS specification:

```
*)
  ENTITY approval
    SUBTYPE OF(relationship);
    approved                 :relationship;
    approver                    :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

approved : the **relationship** that is approved in the **approval**

approver : the **possible_individual** that is the approver in the **approval**

5.2.23.2 class_of_approval

A **class_of_approval** is a **class_of_relationship** whose members are members of **approval** that indicates that members of the **class_of_individual** are approvers in an **approval** for the members of the **class** that are approved.

EXAMPLE That site managers approve design specifications for construction (a **class_of_involvement_by_reference**) is an example of **class_of_approval**.

EXPRESS specification:

```
*)
  ENTITY class_of_approval
    SUBTYPE OF(class_of_relationship);
    class_of_approved           :class_of_relationship;
    class_of_approver           :class_of_individual;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_approved : the **class_of_relationship** whose members are approved by the members of the **class_of_approver**

class_of_approver : the **class_of_individual** whose members are the approvers of the **class_of_relationship** approved

5.2.23.3 class_of_approval_by_status

A **class_of_approval_by_status** is a **class_of_relationship** that indicates a status of the approval that is independent of what is being approved by whom.

EXAMPLE approved, approved with comments, disapproved with comments are examples of **class_of_approval_by_status**.

EXPRESS specification:

```
*)
  ENTITY class_of_approval_by_status
    SUBTYPE OF(class_of_relationship);
  END_ENTITY;
(*
```

5.2.23.4 class_of_lifecycle_stage

A **class_of_lifecycle_stage** is a **class_of_relationship** whose members are members of **lifecycle_stage**.

EXAMPLE Planned, required, expected, and proposed can be represented by instances of **class_of_lifecycle_stage**.

EXPRESS specification:

```
*)
  ENTITY class_of_lifecycle_stage
    SUBTYPE OF(class_of_relationship);
  END_ENTITY;
(*
```

5.2.23.5 lifecycle_stage

A **lifecycle_stage** is a **relationship** that indicates the interest that a **possible_individual** has in some **possible_individual**.

EXAMPLE The relation that links a possible building to a temporal part of the XYZ Corp. can be represented by an instance of **lifecycle_stage**. The nature of that **lifecycle_stage** (e.g. 'planned') can be expressed by classifying with the applicable **class_of_lifecycle_stage**.

EXPRESS specification:

```
*)
  ENTITY lifecycle_stage
    SUBTYPE OF(relationship);
    interest                      :possible_individual;
    interested                    :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

interest : the **possible_individual** that is of interest to the referenced **possible_individual**

interested : the **possible_individual** that has an interest in the referenced **possible_individual**

5.2.24 Possible and intended roles

This subclause contains the declarations of entity data types that represent possible and intended roles.

NOTE Figure 200 is a diagram of the entity data type(s) defined in this subclause (see 4.8.4.8.1).

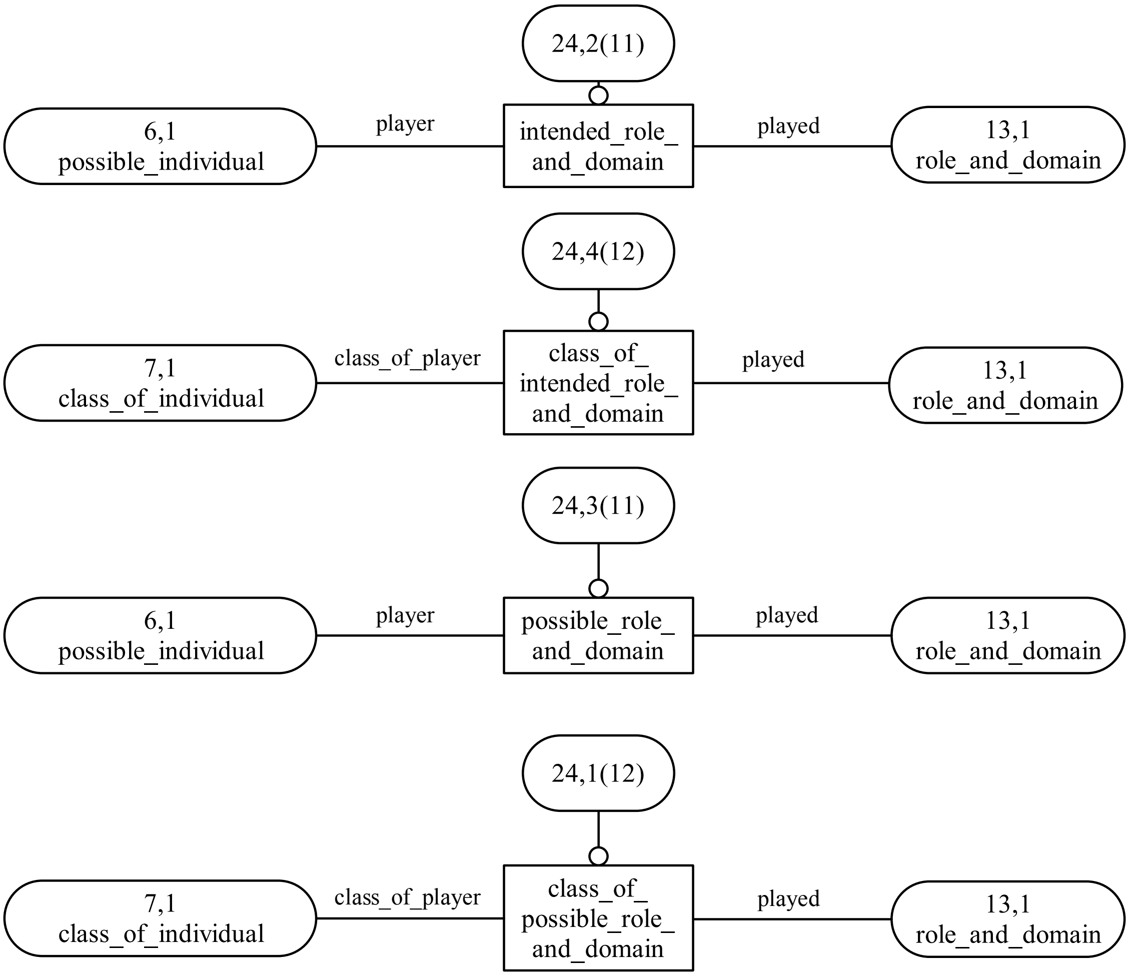


Figure 200 — lifecycle_integration_schema EXPRESS-G diagram 24 of 29

5.2.24.1 class_of_intended_role_and_domain

A **class_of_intended_role_and_domain** is a **class_of_relationship** that indicates that a member of the **class_of_individual** is intended to act as a member of the **role_and_domain**.

EXAMPLE Pumps are intended to play the **role_and_domain** of performer in some pumping activity.

EXPRESS specification:

```
*)
  ENTITY class_of_intended_role_and_domain
    SUBTYPE OF(class_of_relationship);
    class_of_player            :class_of_individual;
    played                     :role_and_domain;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_player : the **class_of_individual** whose members may play the intended **role_and_domain** played : the **role_and_domain** that is intended to be played by members of the **class_of_individual**

5.2.24.2 class_of_possible_role_and_domain

A **class_of_possible_role_and_domain** is a **class_of_relationship** that indicates the **role_and_domain** that can be played by a member of the **class_of_individual**, in some **activity**.

EXAMPLE Pumps can play the **role** of anchor (although they are not intended to do so).

EXPRESS specification:

```
*)
  ENTITY class_of_possible_role_and_domain
    SUBTYPE OF(class_of_relationship);
    class_of_player            :class_of_individual;
    played                     :role_and_domain;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_player : the **class_of_individual** whose members can play the referenced **role_and_domain**

played : the **role_and_domain** that can be played by members of the referenced **class_of_individual**

5.2.24.3 intended_role_and_domain

An **intended_role_and_domain** is a **relationship** that indicates the **role_and_domain** some temporal part of the **possible_individual** is intended to take with respect to some **activity**.

EXAMPLE Some **possible_individual** that is classified as a pump is intended to play the **role_and_domain** of a performer in some pumping activity.

EXPRESS specification:

```
*)
  ENTITY intended_role_and_domain
    SUBTYPE OF(relationship);
    played                     :role_and_domain;
    player                     :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

played : the **role_and_domain** that is intended to be played by the referenced **possible_individual**

player : the **possible_individual** that is intended to play the referenced **role_and_domain**

5.2.24.4 possible_role_and_domain

A **possible_role_and_domain** is a **relationship** that indicates that a player **possible_individual** can possibly play the played **role_and_domain**.

EXAMPLE Acting as an anchor is a possible role for pump 1234.

EXPRESS specification:

```
*)
  ENTITY possible_role_and_domain
    SUBTYPE OF(relationship);
    played                      :role_and_domain;
    player                      :possible_individual;
  END_ENTITY;
(*
```

Attribute definitions:

played	: the **role_and_domain** that the **possible_individual** can play
player	: the **possible_individual** that can play the **role_and_domain**

5.2.25 Set operations

This subclause contains the declarations of entity data types that represent set operations.

NOTE Figure 201 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.5.2.1 and 4.9).

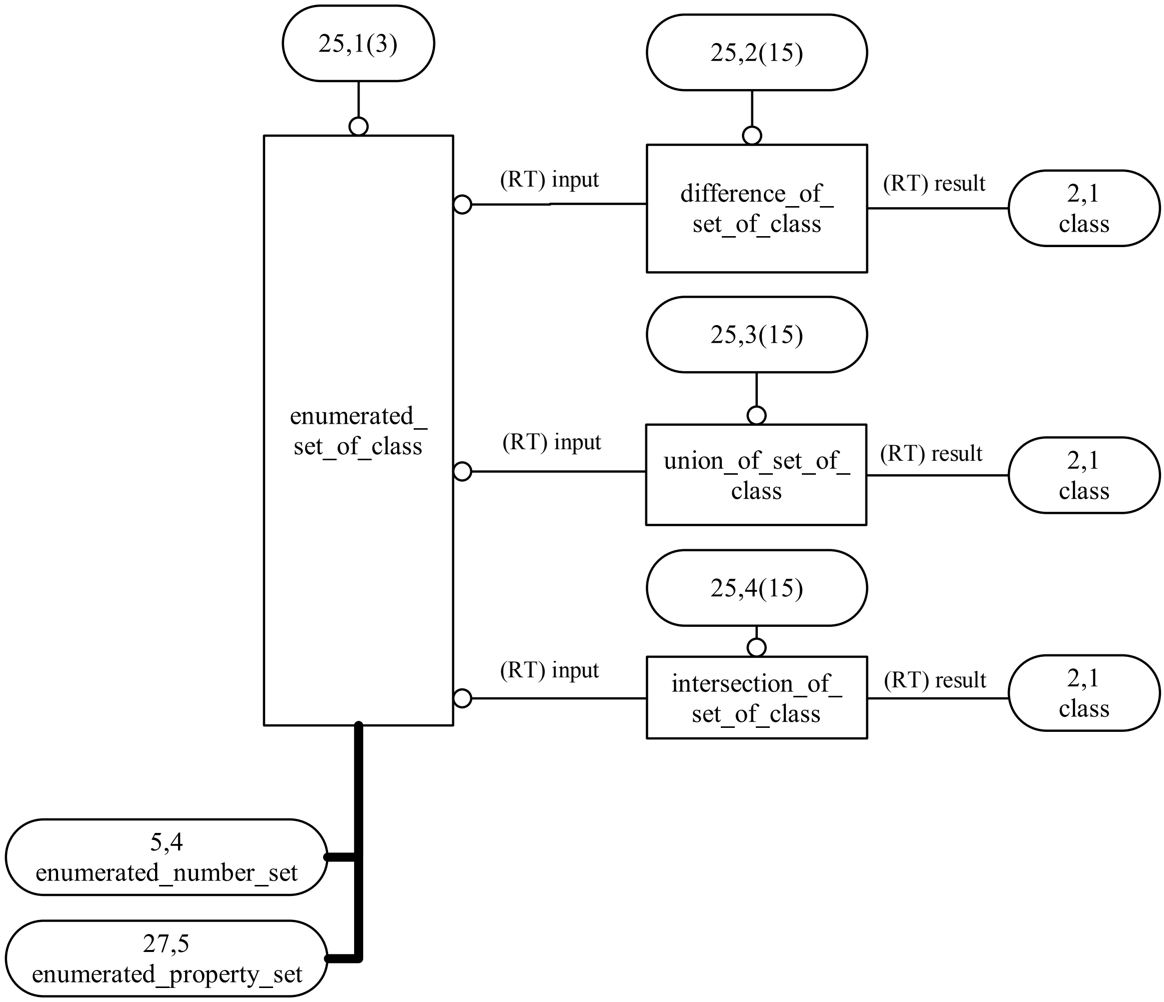


Figure 201 — lifecycle_integration_schema EXPRESS-G diagram 25 of 29

5.2.25.1 difference_of_set_of_class

A **difference_of_set_of_class** is a **functional_mapping** that indicates that the membership of the result **class** is the difference between the membership of the union of the classes that are members of the **enumerated_set_of_class** and their intersection.

NOTE When the **enumerated_set_of_class** consists of a **class** and another **class** that is a subclass of the first class, then the difference is the complement of the subclass.

EXAMPLE The difference of the **enumerated_set_of_class** {{A,B,C},{B,C,D},{C,D,E}} is {A,B,D,E}.

EXPRESS specification:

```
*)
  ENTITY difference_of_set_of_class
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input :     enumerated_set_of_class;
    SELF\functional_mapping.result      :     class;
  END_ENTITY;
 (*
```

Attribute definitions:

input : the **enumerated_set_of_class** that is the domain of the difference function

result : the **class** that is the range of the difference function

5.2.25.2 enumerated_set_of_class

An **enumerated_set_of_class** is a **class_of_class** that is an enumerated set of the instances of **class**. Enumerated means that the full set of members is specified.

EXAMPLE {Plastic, 1.2kg, frame} is an **enumerated_set_of_class**. More generally {{A,B,C},{B,C,D},{C,D,E}} is an **enumerated_set_of_class**. Also "electrical engineering classes for ERDL V1.1" is an **enumerated_set_of_class**.

EXPRESS specification:

```
*)
  ENTITY enumerated_set_of_class
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.25.3 intersection_of_set_of_class

An **intersection_of_set_of_class** is a **functional_mapping** that indicates that the result **class** consists of those members of the members of the classes **enumerated_set_of_class** that are common to each class.

EXAMPLE The intersection of the **enumerated_set_of_class** {{A,B,C},{B,C,D},{C,D,E}} is {C}.

EXPRESS specification:

```
*)
  ENTITY intersection_of_set_of_class
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input :     enumerated_set_of_class;
    SELF\functional_mapping.result      :     class;
  END_ENTITY;
(*
```

Attribute definitions:

input : the **enumerated_set_of_class** whose members are intersected

result : the **class** that represents the intersection of the members of the **enumerated_set_of_class**

5.2.25.4 union_of_set_of_class

A **union_of_set_of_class** is a **functional_mapping** that indicates that the membership of the result **class** is the union of the members of the **enumerated_set_of_class** classes.

EXAMPLE The union of the **enumerated_set_of_class** {{A,B,C},{B,C,D},{C,D,E}} is {A,B,C,D,E}.

EXPRESS specification:

```
*)
  ENTITY union_of_set_of_class
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input :     enumerated_set_of_class;
    SELF\functional_mapping.result      :     class;
  END_ENTITY;
(*
```

Attribute definitions:

input : the **enumerated_set_of_class** that is the domain of the union function

result : the **class** that is the range of the union function

5.2.26 Properties

This subclause contains the declarations of entity data types that represent properties.

NOTE Figure 202 is a diagram of the entity data type(s) defined in this subclause (see 4.8.4.3).

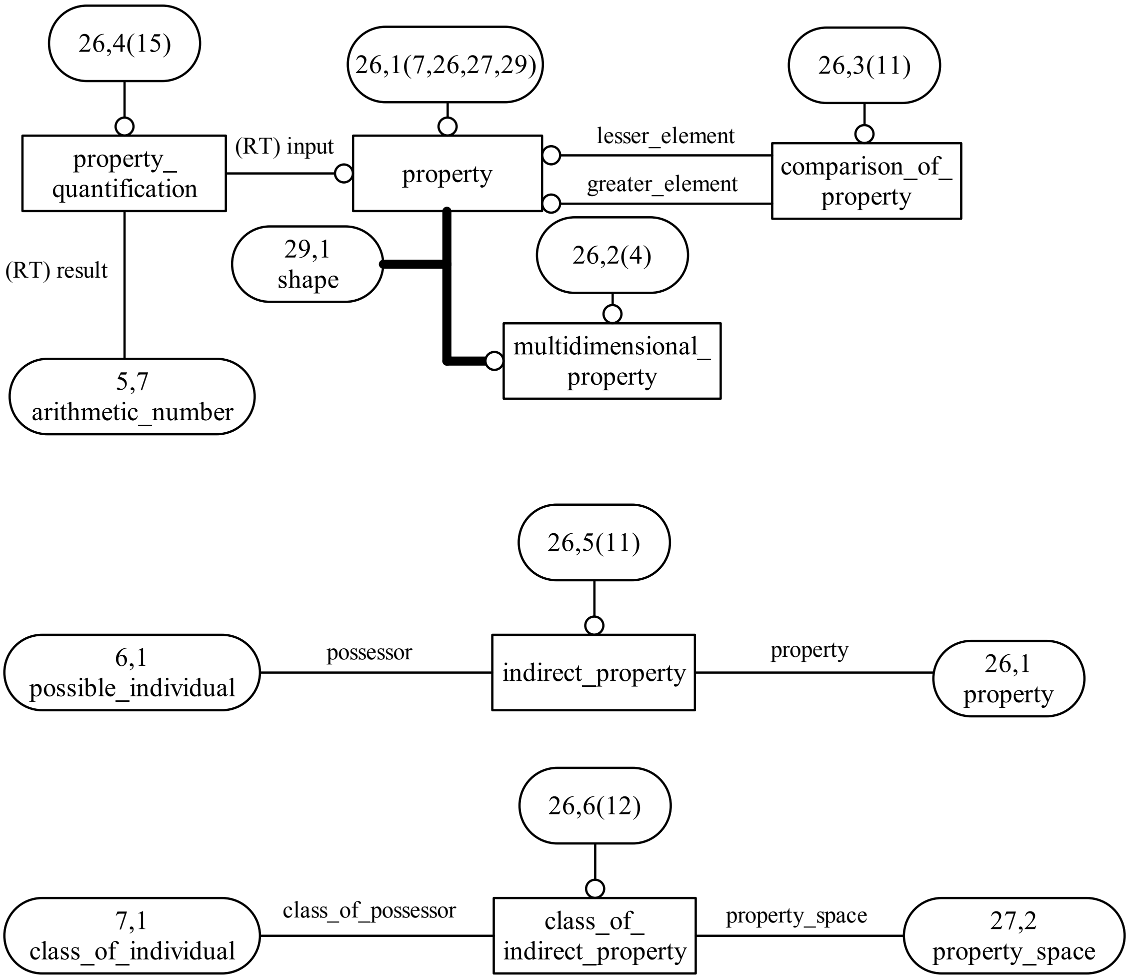


Figure 202 — lifecycle_integration_schema EXPRESS-G diagram 26 of 29

5.2.26.1 class_of_indirect_property

A **class_of_indirect_property** is a **class_of_relationship** that indicates that a member of the **class_of_individual** can possess a member of the **class_of_property** as an **indirect_property** of this type.

EXAMPLE Maximum Allowable Working Pressure is a **class_of_indirect_property** that is indicated by a pressure, and can be possessed by a pressure vessel.

EXPRESS specification:

```
*)
  ENTITY class_of_indirect_property
    SUBTYPE OF(class_of_relationship);
    class_of_possessor         :class_of_individual;
    property_space             :property_space;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_possessor : the **class_of_individual** whose instances may possess a member of the **property_space**

property_space : the **property_space** a member of which may be possessed by a member of the **class_of_individual**

5.2.26.2 comparison_of_property

A **comparison_of_property** is a **relationship** that indicates the magnitude of one **property** is greater than that of another.

EXAMPLE That the temperature in a room is less than that in a furnace can be indicated by an instance of **comparison_of_property**.

EXPRESS specification:

```
*)
  ENTITY comparison_of_property
    SUBTYPE OF(relationship);
    greater_element            :property;
    lesser_element             :property;
  END_ENTITY;
(*
```

Attribute definitions:

greater_element : the **property** that is the greater element in a **comparison_of_property**

lesser_element : the **property** that is the lesser element in the **comparison_of_property**

5.2.26.3 indirect_property

An **indirect_property** is a **relationship** between a **property** and a **possible_individual**. The nature of the **indirect_property** is defined by its **classification** by a **class_of_indirect_property**.

A property is indirect when it does not apply directly to the **possible_individual** to which it applies, but is derived from some process.

NOTE A property is indirect because it does not directly apply. There can only be one temperature that a thing has (at a time), so a Maximum Allowable Working Temperature is not its temperature, but an indirect property derived from doing some tests or calculations to determine its value (as opposed to it being a current measurement). This is what makes it indirect.

EXAMPLE A Maximum Allowable Working Pressure of 50 BarA for V101 is specified by an **indirect_property** between the pressure of 50 BarA and V101, classified by the **class_of_indirect_property** Maximum Allowable Working Pressure.

EXPRESS specification:

```
*)
  ENTITY indirect_property
    SUBTYPE OF(relationship);
    possessor                   :possible_individual;
    property                    :property;
  END_ENTITY;
(*
```

Attribute definitions:

possessor : the **possible_individual** that possesses the **indirect_property**

property : the **property** that is indirectly possessed by the **possible_individual**

5.2.26.4 multidimensional_property

A **multidimensional_property** is a **property** that is also a **multidimensional_object**.

EXAMPLE A pump flow head characteristic is a **multidimensional_object**. It consists of a continuum of Q, H property pairs, where Q is the flow rate and H is the flowing head difference. Each pair of properties Q_a and H_a, where Q_a is a particular flow rate and H_a a particular head, is a **multidimensional_property** [Q_a, H_a].

EXPRESS specification:

```
*)
  ENTITY multidimensional_property
    SUBTYPE OF(property, multidimensional_object);
  END_ENTITY;
(*
```

5.2.26.5 property

A **property** is a **class_of_individual** that is a member of a continuum of a **class_of_property**. The **property** may be quantified by mapping to a number on a scale.

NOTE 1 A member of a **property** is a **possible_individual** that has the same degree or magnitude of the quality or characteristic represented by the **property** as other members.

NOTE 2 The types of characteristic or quality, such as temperature or density, are instances of **class_of_property**.

NOTE 3 Duplicate properties (e.g. that map to the same number on the same scale) should not be created within the same data store.

EXAMPLE A particular degree of hotness can be represented as an instance of **property**.

EXPRESS specification:

```
*)
  ENTITY property
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.26.6 property_quantification

A **property_quantification** is a **functional_mapping** whose members map a **property** to an **arithmetic_number**.

EXAMPLE The link that maps a particular mass to the number 4.2 can be represented by an instance of **property_quantification**.

NOTE 1 The actual representation of the number is done by linking the **arithmetic_number** to a **class_of_EXPRESS_information_representation** via a **class_of_representation_of_thing**.

NOTE 2 The unit or scale of the quantification is given by classifying the **property_quantification** by a **scale**.

EXPRESS specification:

```
*)
  ENTITY property_quantification
    SUBTYPE OF(functional_mapping);
    SELF\functional_mapping.input       :     property;
    SELF\functional_mapping.result      :     arithmetic_number;
  END_ENTITY;
(*
```

Attribute definitions:

input : the **property** that is quantified by the referenced **arithmetic_number**

result : the **arithmetic_number** that quantifies the referenced **property**

5.2.27 Classes of property

This subclause contains the declarations of entity data types that represent classes of property.

NOTE Figure 203 is a diagram of the entity data type(s) defined in this subclause (see 4.8.4.3).

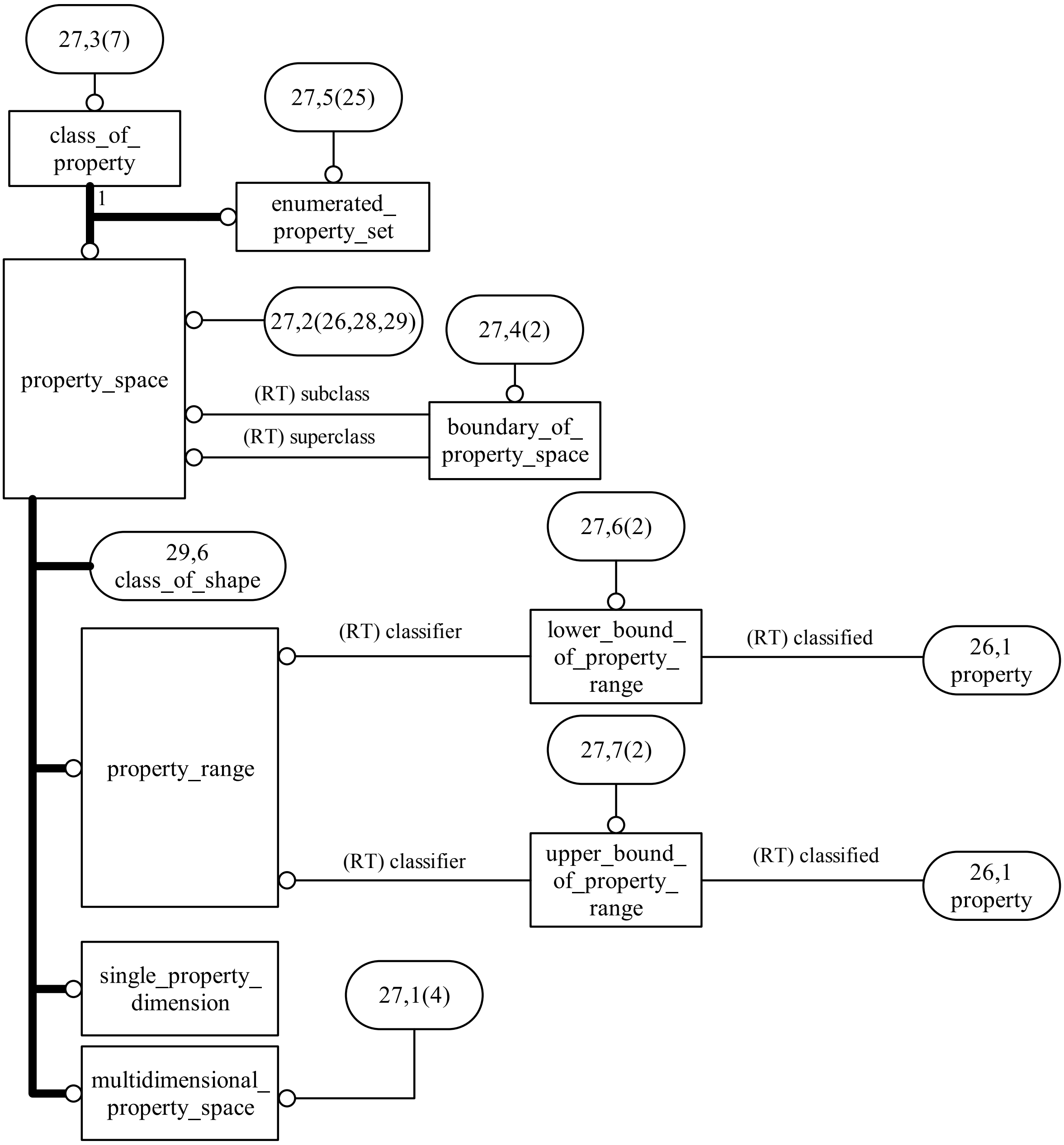


Figure 203 — lifecycle_integration_schema EXPRESS-G diagram 27 of 29

5.2.27.1 boundary_of_property_space

A **boundary_of_property_space** is a **specialization** that indicates the members of the subclass form a boundary of the superclass.

EXAMPLE The **property_space** that corresponds to the maximum speed head flow curve is a boundary of the **property_space** that corresponds to the pump operating envelope.

EXPRESS specification:

```
*)
  ENTITY boundary_of_property_space
```

```
    SUBTYPE OF(specialization);
    SELF\specialization.subclass: property_space;
    SELF\specialization.superclass      :     property_space;
  END_ENTITY;
(*
```

Attribute definitions:

subclass : the **property_space** whose members form the boundary of the **property_space** referenced by the superclass attribute

superclass : the **property_space** that is bounded by the members of the **property_space** referenced by the subclass attribute

5.2.27.2 class_of_property

A **class_of_property** is a **class_of_class_of_individual** whose members are instances of **property**.

EXAMPLE 'Temperature' is an example of **class_of_property**.

EXPRESS specification:

```
*)
  ENTITY class_of_property
    SUPERTYPE OF (ONEOF(property_space, enumerated_property_set))
    SUBTYPE OF(class_of_class_of_individual);
  END_ENTITY;
(*
```

5.2.27.3 enumerated_property_set

An **enumerated_property_set** is a **class_of_property** and an **enumerated_set_of_class** whose members are an enumerated set of properties of the same **single_property_dimension** or **multidimensional_property_space**.

EXAMPLE {115 Volt, 240 Volt} is an example of an **enumerated_property_set**.

EXPRESS specification:

```
*)
  ENTITY enumerated_property_set
    SUBTYPE OF(class_of_property, enumerated_set_of_class);
  END_ENTITY;
(*
```

5.2.27.4 lower_bound_of_property_range

A **lower_bound_of_property_range** is a **class_of_classification** that indicates that a **property** is the lower bound of a **property_range**.

EXAMPLE -10 Celsius is the lower bound of the range -10 to +20 Celsius.

EXPRESS specification:

```
*)
  ENTITY lower_bound_of_property_range
    SUBTYPE OF(classification);
    SELF\classification.classified      :     property;
    SELF\classification.classifier      :     property_range;
  END_ENTITY;
(*
```

Attribute definitions:

classified : the **property** that as classified is the lower bound in the **lower_bound_of_property_range**

classifier : the **property_range** that is bounded as classifier in the **lower_bound_of_property_range**

5.2.27.5 multidimensional_property_space

A **multidimensional_property_space** is a **property_space** and a **multidimensional_object** whose members are properties each of which maps to more than one number. Each property will consist of elements of the same property dimensions.

EXAMPLE A pump performance curve of flowrate and differential head is a **multidimensional_property_space**.

EXPRESS specification:

```
*)
  ENTITY multidimensional_property_space
    SUBTYPE OF(property_space, multidimensional_object);
  END_ENTITY;
(*
```

5.2.27.6 property_range

A **property_range** is a **property_space** that is a continuous subset of a **single_property_dimension**.

EXAMPLE -10C to +20C is a **property_range** of temperature.

EXPRESS specification:

```
*)
  ENTITY property_range
    SUBTYPE OF(property_space);
  END_ENTITY;
(*
```

5.2.27.7 property_space

A **property_space** is a **class_of_property** whose members are a coherent continuum of **property**.

EXAMPLE 1 The set of temperature properties, known as temperature, is a **property_space**.

EXAMPLE 2 The members of the pressure and flow rate **class_of_property** that fall on a particular pump curve is a **property_space**.

EXPRESS specification:

```
*)
  ENTITY property_space
    SUBTYPE OF(class_of_property);
  END_ENTITY;
(*
```

5.2.27.8 single_property_dimension

A **single_property_dimension** is a **property_space** that is a single and complete continuum of properties each of which maps to a single number.

EXAMPLE Temperature, pressure, viscosity, and length are examples of **single_property_dimension**.

EXPRESS specification:

```
*)
  ENTITY single_property_dimension
    SUBTYPE OF(property_space);
  END_ENTITY;
(*
```

5.2.27.9 upper_bound_of_property_range

An **upper_bound_of_property_range** is a **class_of_classification** that indicates that the **property** is the upper bound of the **property_range**.

EXAMPLE +20 Celsius is the upper bound of the range -10 to +20 Celsius.

EXPRESS specification:

```
*)
  ENTITY upper_bound_of_property_range
    SUBTYPE OF(classification);
    SELF\classification.classified      :     property;
    SELF\classification.classifier      :     property_range;
  END_ENTITY;
(*
```

Attribute definitions:

classified : the **property** that as the classified in the upper bound in the **upper_bound_of_property_range**

classifier : the **property_range** that as the classifier has an upper bound specified in the **upper_bound_of_property_range**

5.2.28 Scale conversions

This subclause contains the declarations of entity data types that represent scale conversions.

NOTE Figure 204 is a diagram of the entity data type(s) defined in this subclause (see also 4.8.4.3.2).

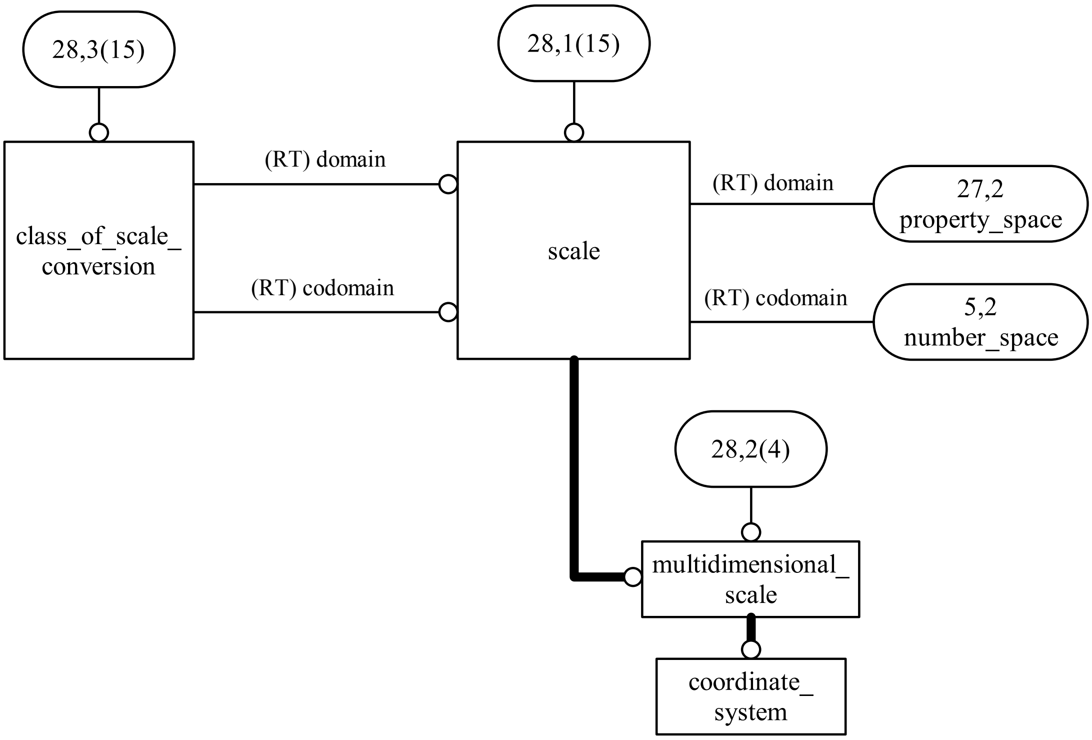


Figure 204 — lifecycle_integration_schema EXPRESS-G diagram 28 of 29

5.2.28.1 class_of_scale_conversion

A **class_of_scale_conversion** is a **class_of_isomorphic_functional_mapping** that defines a conversion between two different scales of units used for the quantification of properties.

EXAMPLE The Fahrenheit scale for temperature and the Celsius scale for temperature can each be represented by instances of **scale**. The conversion between these scales can be represented by an instance of **class_of_scale_conversion**.

EXPRESS specification:

```
*)
  ENTITY class_of_scale_conversion
    SUBTYPE OF(class_of_isomorphic_functional_mapping);
    SELF\class_of_functional_mapping.codomain :     scale;
    SELF\class_of_functional_mapping.domain   :     scale;
  END_ENTITY;
(*
```

Attribute definitions:

codomain : the second **scale** for which the conversion is asserted

domain : the first **scale** for which the conversion is asserted

5.2.28.2 coordinate_system

A **coordinate_system** is a **multidimensional_scale** for locating and relating a **possible_individual** in an n-dimensional space in which arbitrary geometric transformations are valid.

EXAMPLE The XYZ site coordinate system, is an example of a **coordinate_system**.

EXPRESS specification:

```
*)
  ENTITY coordinate_system
    SUBTYPE OF(multidimensional_scale);
  END_ENTITY;
(*
```

5.2.28.3 multidimensional_scale

A **multidimensional_scale** is a **scale** that is also a **multidimensional_object**.

EXAMPLE A [Celsius, seconds] scale is a **multidimensional_scale** on which temperature variation over time can be plotted.

EXPRESS specification:

```
*)
  ENTITY multidimensional_scale
    SUBTYPE OF(scale, multidimensional_object);
  END_ENTITY;
(*
```

5.2.28.4 scale

A **scale** is a **class_of_isomorphic_functional_mapping** whose members are members of **property_quantification**. It indicates the **number_space** a **property_space** maps to for the **scale** in question.

EXAMPLE The link that is known as the Celsius scale between the **class_of_number** [-273, inf] and the **class_of_property** temperature can be represented by an instance of **scale**.

EXPRESS specification:

```
*)
  ENTITY scale
    SUBTYPE OF(class_of_isomorphic_functional_mapping);
    SELF\class_of_functional_mapping.codomain :     number_space;
    SELF\class_of_functional_mapping.domain   :     property_space;
  END_ENTITY;
(*
```

Attribute definitions:

codomain : the **number_space** whose members can quantify the members of the referenced **property_space**

domain : the **class_of_property** whose members can be quantified by members of the referenced **class_of_number**

5.2.29 Shapes

This subclause contains the declarations of entity data types that represent shapes.

NOTE Figure 205 is a diagram of the entity data type(s) defined in this subclause (see 4.8.4.5).

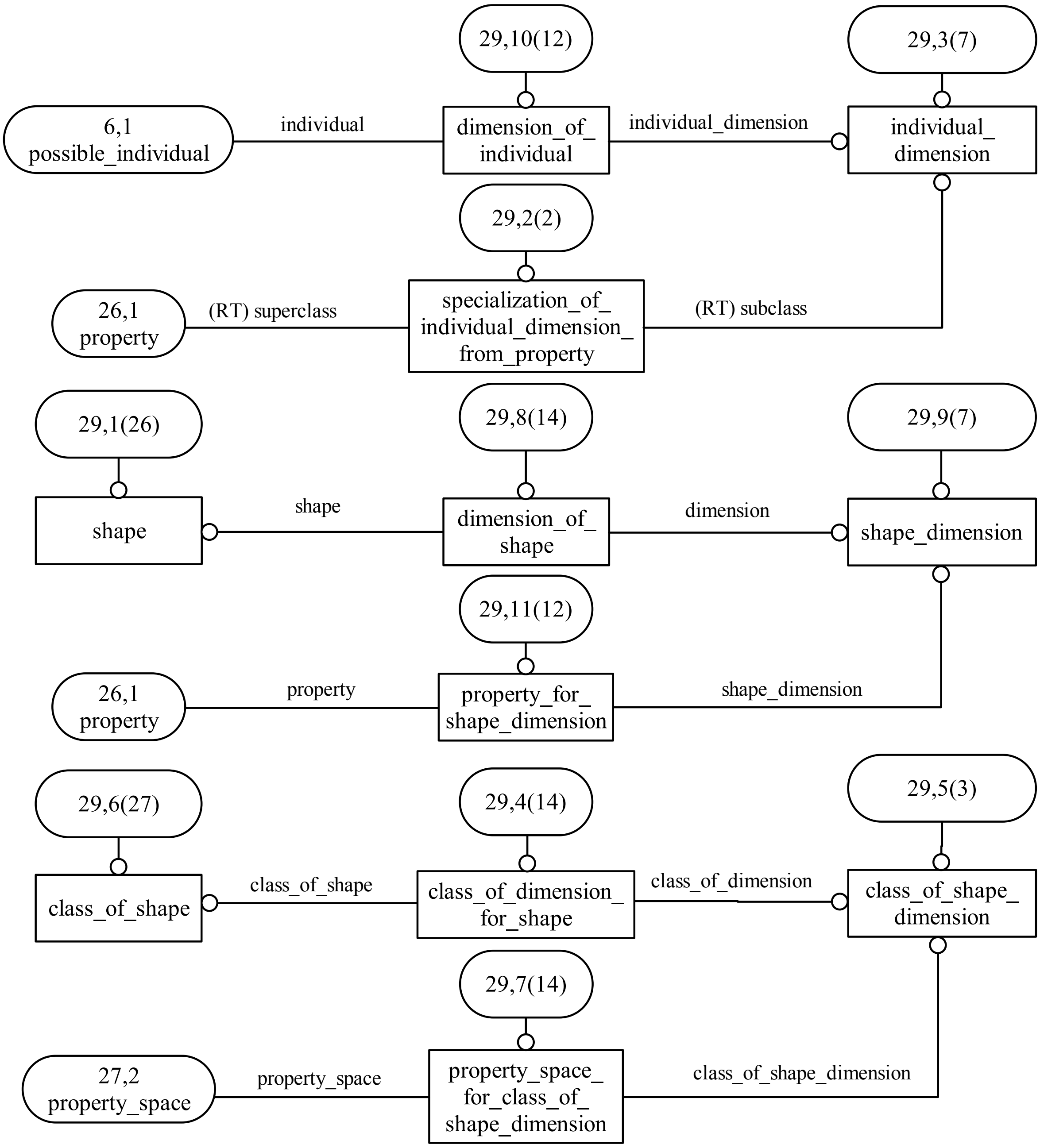


Figure 205 — lifecycle_integration_schema EXPRESS-G diagram 29 of 29

5.2.29.1 class_of_dimension_for_shape

A **class_of_dimension_for_shape** is a **class_of_class_of_relationship** that indicates that members of the class_of_shape have a dimension that is a member of the class_of_dimension.

EXAMPLE Specifying that members of the "class of circle" have members of "class of diameter" is an instance of **class_of_dimension_for_shape**.

EXPRESS specification:

```
*)
  ENTITY class_of_dimension_for_shape
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_dimension          :class_of_shape_dimension;
    class_of_shape              :class_of_shape;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_dimension : the **class_of_shape_dimension** in the **class_of_dimension_for_shape**

class_of_shape : the **class_of_shape** in the **class_of_dimension_for_shape**

5.2.29.2 class_of_shape

A **class_of_shape** is a **property_space** that has instances of **shape** as its members.

EXAMPLE Regular geometric forms such as line, circle, square, cylinder, sphere, and cone can be represented by instances of **class_of_shape**.

EXPRESS specification:

```
*)
  ENTITY class_of_shape
    SUBTYPE OF(property_space);
  END_ENTITY;
(*
```

5.2.29.3 class_of_shape_dimension

A **class_of_shape_dimension** is a **class_of_class** that is a dimension of a **class_of_shape**.

EXAMPLE Diameter, height, and width (in general rather than a particular one) are examples of **class_of_shape_dimension**.

EXPRESS specification:

```
*)
  ENTITY class_of_shape_dimension
    SUBTYPE OF(class_of_class);
  END_ENTITY;
(*
```

5.2.29.4 dimension_of_individual

A **dimension_of_individual** is a **class_of_relationship** that indicates that each member of the set of lines that are the **individual_dimension** are a dimension of the **possible_individual**.

EXAMPLE The set of all lines that pass through the centre of a particular circle and end at the circumference of that circle, are a dimension (diameter) of that circle. The particular dimension is indicated by the **dimension_of_shape** that classifies the **dimension_of_individual**.

EXPRESS specification:

```
*)
  ENTITY dimension_of_individual
    SUBTYPE OF(class_of_relationship);
    individual                  :possible_individual;
    individual_dimension        :individual_dimension;
  END_ENTITY;
(*
```

Attribute definitions:

individual : the **possible_individual** that is assigned an **individual_dimension** in the **dimension_of_individual**

individual_dimension : the **individual_dimension** for the **possible_individual** in the **dimension_of_individual**

5.2.29.5 dimension_of_shape

A **dimension_of_shape** is a **class_of_class_of_relationship** that indicates that members of the **shape_dimension** are dimensions of the **shape** members.

EXAMPLE The sets of 10m lines that are diameters of 10m circles is an example of **dimension_of_shape**.

EXPRESS specification:

```
*)
  ENTITY dimension_of_shape
    SUBTYPE OF(class_of_class_of_relationship);
    dimension                   :shape_dimension;
    shape                       :shape;
  END_ENTITY;
(*
```

Attribute definitions:

dimension : the **shape_dimension** of the **shape**

shape : the **shape** that possesses the **shape_dimension**

5.2.29.6 individual_dimension

An **individual_dimension** is a **class_of_individual** whose members characterize a particular **possible_individual**.

EXAMPLE The set of lines that are each a diameter of a particular circle.

EXPRESS specification:

```
*)
  ENTITY individual_dimension
    SUBTYPE OF(class_of_individual);
  END_ENTITY;
(*
```

5.2.29.7 property_for_shape_dimension

A **property_for_shape_dimension** is a **class_of_class_of_relationship** that indicates that the members of the **shape_dimension** are of the **property**.

EXAMPLE 10m diameter is a 10m length.

EXPRESS specification:

```
*)
  ENTITY property_for_shape_dimension
    SUBTYPE OF(class_of_relationship);
    property                    :property;
    shape_dimension             :shape_dimension;
  END_ENTITY;
(*
```

Attribute definitions:

property : the **property** for the **property_for_shape_dimension**

shape_dimension : the **shape_dimension** whose members are specializations of the **property** in the **property_for_shape_dimension**

5.2.29.8 property_space_for_class_of_shape_dimension

A **property_space_for_class_of_shape_dimension** is a **class_of_class_of_relationship** that indicates the **property_space** that a **class_of_shape_dimension** is from.

EXAMPLE Diameter is a length dimension.

EXPRESS specification:

```
*)
  ENTITY property_space_for_class_of_shape_dimension
    SUBTYPE OF(class_of_class_of_relationship);
    class_of_shape_dimension    :class_of_shape_dimension;
    property_space              :property_space;
  END_ENTITY;
(*
```

Attribute definitions:

class_of_shape_dimension : the **class_of_shape_dimension** whose members have a **property** in the referenced **property_space**

property_space : the **property_space** that the **class_of_shape_dimension** is from

5.2.29.9 shape

A **shape** is a **property** that depends on constant relations of position and proportionate distance among all the points composing its outline or its external surface.

EXAMPLE 1 20mm diameter circle and 10-20mm diameter circles are examples of **shape**.

EXAMPLE 2 Irregular forms such as the outer envelope of a model of pump can be represented by instances of **shape**.

EXPRESS specification:

```
*)
  ENTITY shape
    SUBTYPE OF(property);
  END_ENTITY;
(*
```

5.2.29.10 shape_dimension

A **shape_dimension** is a **class_of_class_of_individual** that is a set of **individual_dimension** that define an aspect of a shape.

EXAMPLE Diameter of 5m, height of 3mm, and width of 10cm are members of **shape_dimension**.

EXPRESS specification:

```
*)
  ENTITY shape_dimension
    SUBTYPE OF(class_of_class_of_individual);
  END_ENTITY;
(*
```

5.2.29.11 specialization_of_individual_dimension_from_property

A **specialization_of_individual_dimension_from_property** is a **specialization** that indicates the members of the dimension are members of the property.

EXAMPLE A diameter of 10m is a length of 10m.

EXPRESS specification:

```
*)
  ENTITY specialization_of_individual_dimension_from_property
    SUBTYPE OF(specialization);
    SELF\specialization.subclass: individual_dimension;
    SELF\specialization.superclass      :    property;
  END_ENTITY;
(*
```

Attribute definitions:

subclass : the **individual_dimension** that is the specialization

superclass : the **property** that is the generalization

```
*)

END_SCHEMA;
```

Annex A
(normative)

Information object registration

A.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

```
{iso standard 15926 part{2} version{1}}
```

is assigned to this part of ISO 15926. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 15926-1.

NOTE This is the object identifier that will apply to the published (IS) version of this part.

A.2 Schema identification

To provide for unambiguous identification of the process_plant_lifecycle_schema in an open information system, the object identifier

```
{ iso standard 15926 part(2) version(1) object(1) lifecycle-integration-schema(1) }
```

is assigned to the **lifecycle_integration_schema** (see clause 5). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

NOTE This is the object identifier that will apply to the published (IS) version of this schema.

Annex B
(informative)

Computer interpretable listings

This annex references a listing of the EXPRESS schema specified in this part of ISO 15926, without comments or other explanatory text. This listing is available in computer-interpretable form from the Internet:

`http://www.tc184-sc4.org/EXPRESS/`

If there is difficulty accessing this site contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: `sc4sec@tc184-sc4.org`.

NOTE The information provided in computer-interpretable form at the above URL is informative. The information that is contained in the body of this part of ISO 15926 is normative.

Annex C
(informative)

Use of ISO 10303-11 EXPRESS

The data model defined in clause 5 is specified using the EXPRESS language defined in ISO 10303-11. However, not all features of this language are used. The following EXPRESS constructs are excluded from the specification of the data model:

— constructed data types;

— generalized data types;

— select data types;

— parameter data types;

— array and bag aggregate types;

— derived attributes;

— inverse attributes

— domain rules (where clause);

— global rules;

— algorithms;

— constants.

Entity data type and attribute names conform to ISO 10303-11 with no additional restrictions.

Annex D
(informative)

Some notes on set theory in ISO15926

D.1 Introduction

This annex explains some of the principles of set theory that are supported by this part of ISO15926.

D.1.1 What is a set?

A set is a thing that has members, and is defined by its membership (the null set is the set that has no members). That is, if two sets have the same members, they are the same set. If they have different members, they are different sets. Whilst a set is defined by its members, it may be that at any point in time, not all the members of a set are known.

D.1.2 Sets and ISO15926

The paradigm used in ISO15926 of spatio-temporal extents means that objects are unchanging, since change is handled through **possible_individual**s that are temporal parts of the **whole_life_individual** that they represent a state of. This gives the model an "outside of time" viewpoint, rather than a "present" viewpoint and means that all things are unchanging.

So for instance, instead of having a car that at one time is red, and at another time is blue (meaning that the car has changed its set membership) you have two states of the car, one of which is red, and one of which is blue. These set memberships do not change depending on when you are looking at them, from past, present or future. Even when you look forwards from the start of the cars life, the future state of it is still going to be blue; it is just that it is quite likely you do not know this.

D.2 Some different sorts of set theory

A set is a thing that has members, and a set is defined by its membership (the null set is the set that has no members). If two sets have the same members, they are the same set. If they have different members, they are different sets. Whilst a set is defined by its members, it may be that at any point in time, not all the members of a set are known.

D.2.1 Single-level sets

Single-level sets is a system where there are sets and members of sets, but the sets cannot themselves be members of sets. This situation pertains with Entity-Relationship models where entity types cannot be members of other entity types. This is illustrated in Figure D.1.

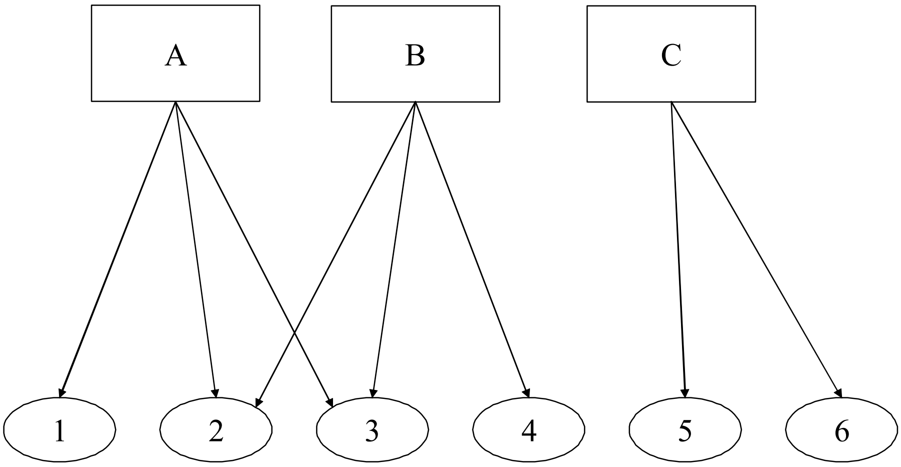


Figure D.1 — Single level sets.

In some cases, it is not even allowed that members be members of more than one set.

D.2.2 Hierarchical sets

With hierarchical sets the situation is that sets at one level may be members of sets at the level above, but there is no crossing of levels. Figure D.2 illustrates this.

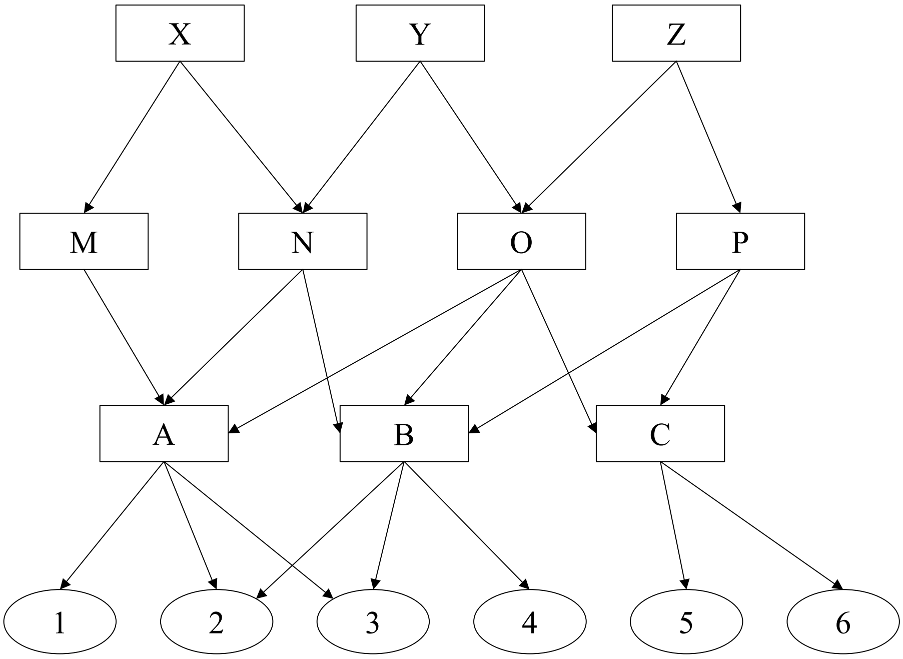


Figure D.2 — An example of hierarchical sets

An example of hierarchical sets in use is in data model, meta-model, meta-meta-model approaches. Hierarchical sets occur naturally, and can be found in this part of ISO15926 through entity types like **individua**l, **class_of_individual**, and **class_of_class_of_individual**.

D.2.3 Well-founded sets

Well-founded sets are the sets of "standard" set theories such as Zermelo-Fraenkel (ZF) set theory, or von Neuman, Bernays, Goedel (VNBG) set theory that can be found in standard texts. Well-founded sets can take members from any level below their own, but are not allowed membership loops (e.g. a set being a member of itself). Figure D.3 shows an example of a system of well-founded sets.

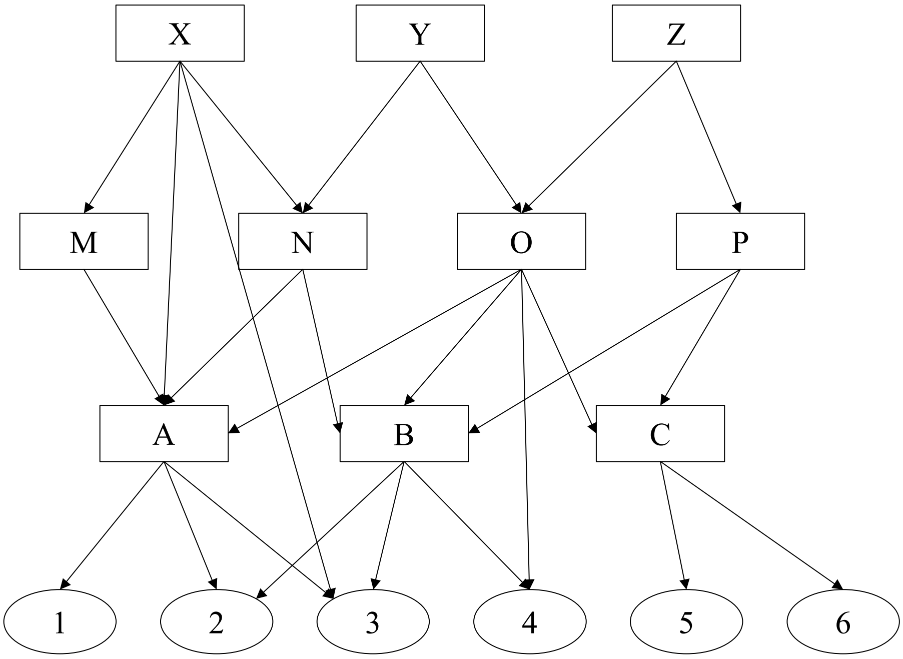


Figure D.3 — An example of well-founded sets.

This form of set theory was largely developed as a reaction (perhaps even an over-reaction) to Russell's Paradox which showed that if sets could be members of themselves, then in certain circumstances paradoxes could arise, such as the set of all sets that do not contain themselves, which cannot exist.

D.2.4 Non-well-founded set theory

The essence of non-well-founded set theory is to allow sets to be members of themselves, where the membership graphs can be constructed. This is illustrated in Figure D.4.

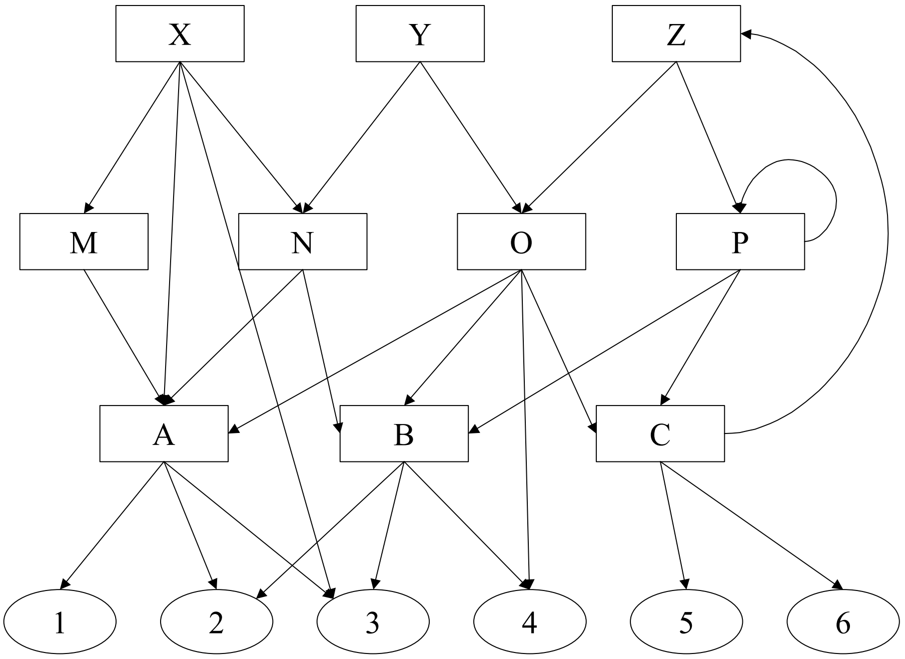


Figure D.4 — An example of non-well-founded sets.

By insisting that sets can be constructed avoids Russell's Paradox, but allows useful things to be said that well-founded sets prevent, like "class is a class", "thing is a class", and "class is a thing".

D.3 Commentary

It should be noted that each type of set is a subset of the following types of set, and since ISO15926 allows non-well-founded sets, it supports all the other types of set mentioned here as well.

Annex E
(informative)

An analysis of the uses and meanings of associations

E.1 Introduction

Historically many data models have taken a snapshot view of the world, which means that when change takes place, history is lost because it is overwritten. EPISTLE (the European Process Industries STEP Technical Liaison Executive) has been concerned with developing data models that are capable of managing information about process plants throughout the life of the plant. This means being able to hold information about the past, the present and the future.

The EPISTLE Core Model (ECM) has gone through a number of versions. In the early versions, history was supported by the use of associations. An association is a way of holding historical information about a relationship that can change over time. The relationship (in the entity-relationship sense) is replaced by an entity type that includes attributes that give the start date and end date for the validity of the relationship. This type of entity type is called an association. This approach is described in [5].

More recently, the way that change is handled in the EPISTLE Core Model has itself been changed. The approach now taken is to manage change through recognising different states of individual things that are valid for a period of time, together with timeless relationships between these. An outline of this approach is presented in [6]. This approach is taken in the conceptual model in clause 5. The reasons for this change are beyond the scope of this annex. However, this annex does look at how the two approaches relate to each other. In particular we have found four basic patterns for how associations are translated into states and timeless relationships, and an approach to analysing associations is presented.

NOTE The names of entity types used in this paper are not necessarily those to be found in clause 5 or the EPISTLE Core Model.

E.2 From snapshot relationships to associations

In Figure E.1, two ways of modelling the ownership of a physical object by an organisation are presented. The first models ownership as a relationship between an organisation and physical object. The problem comes when ownership changes. Either the change is not allowed, or the existing ownership attribute on the physical object is overwritten. This means that the previous ownership history is lost.

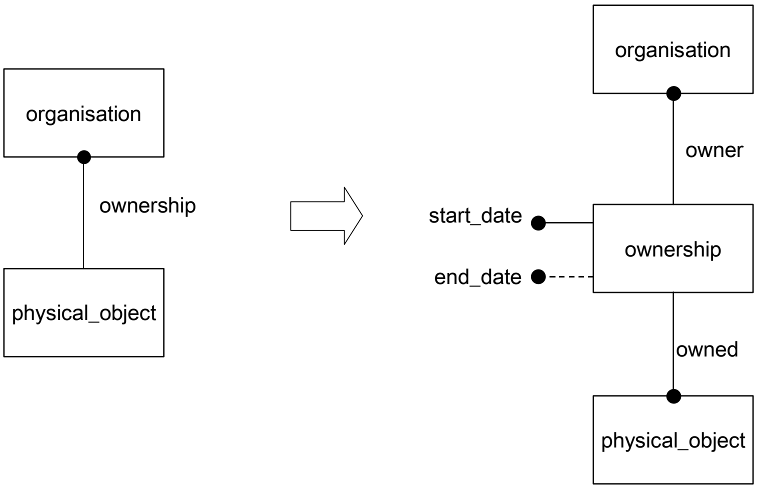


Figure E.1 — Moving from a snapshot model to one using associations.

The second models ownership as an association entity type. This has a start date and an end date as attributes, as well as references to what is owned, and the owner. Now when the ownership changes, the end date of the previous ownership relation gets an end date, and a new ownership record is created. As a result the ownership history can be held.

E.3 From associations to relationships between states and classes

Whilst using associations seems to solve the problem, further analysis suggests that some information is being hidden. In order to uncover this we need to consider individual things as spatio-temporal extents, and states as temporal parts of these spatio-temporal extents for which some relationship is true. To help in this we use space-time maps as illustrated in Figure E.2.

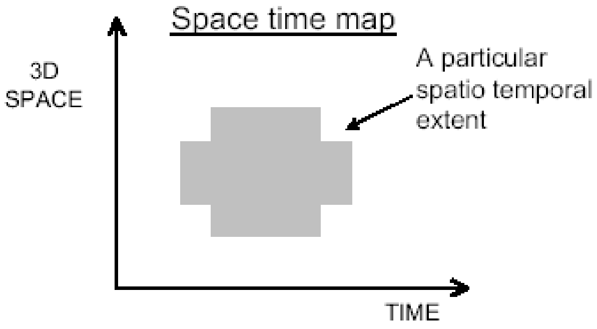


Figure E.2 — Space-time map

Five different patterns have been found of what different sorts of association represent in spatio-temporal terms. These are illustrated in the following subsections.

E.3.1 Pattern 1: a relationship between a state of an individual and a class

Figure E.3 shows a classification pattern. An example is given of where an individual_state, "Car1", is classified as being "Red" from 1/1/2001 to 4/3/2001.

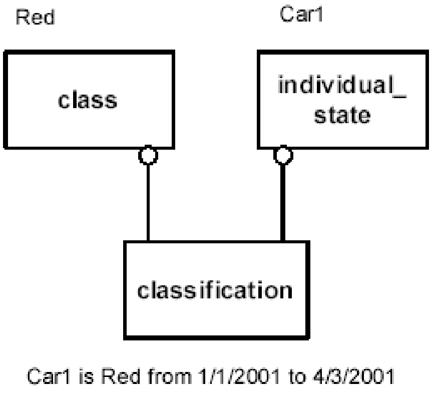


Figure E.3 — An example of a classification association.

If we examine what is happening here using a space-time map, Figure E.4, we see that there is a state of Car1 that is classified as being red (the shaded area).

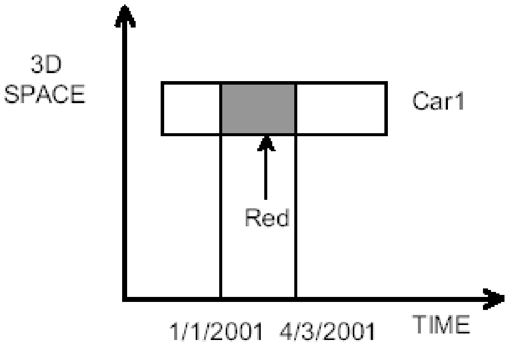


Figure E.4 — A space-time map for classification of an individual.

A data model that represents this space-time diagram is shown in Figure E.5.

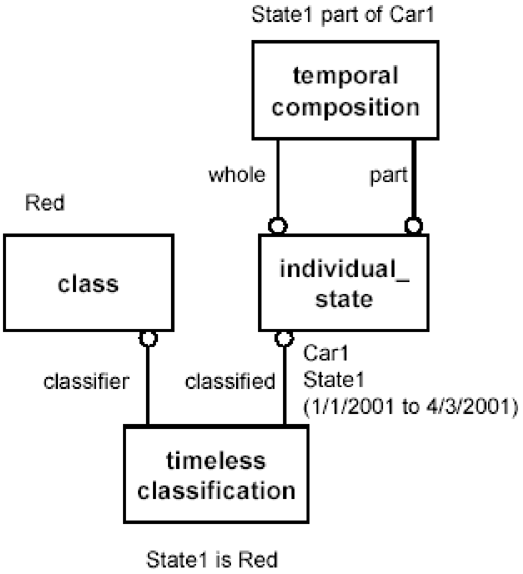


Figure E.5 — Classification using states.

Here the state of the car is modelled explicitly, rather than being part of the classification association, and it is shown as being a temporal part of the whole car, and as being classified as red. The classification relationship is now timeless, because the period of being red is explicit in the state that is classified.

E.3.2 Pattern 2: a relationship between two states of an individual

Figure E.6 below illustrates the case where an association represents a relationship between two states of individual things. To illustrate the model an example is given of how a wheel, "Wheel1", is part of a car, "Car1" from 1/1/2001 to 5/4/2001.

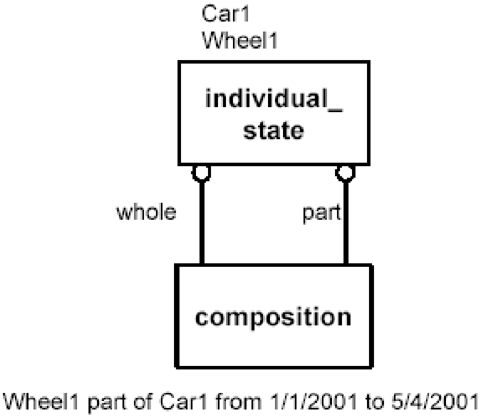


Figure E.6 — Association between two individuals.

Figure E.7 shows this example as a space-time map, showing the different states of the car and wheel, as well as the whole life of the car and wheel.

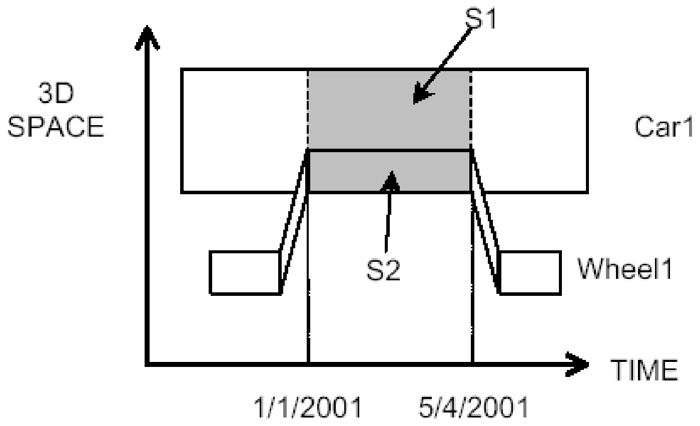


Figure E.7 — A space-time map for composition.

The diagram shows that in this case there is a state of Car1, S1, and a state of the Wheel1, S2, both with the same state and end date, and S2 is a part of S1. When this space-time map is modelled explicitly, the result is found in Figure E.8.

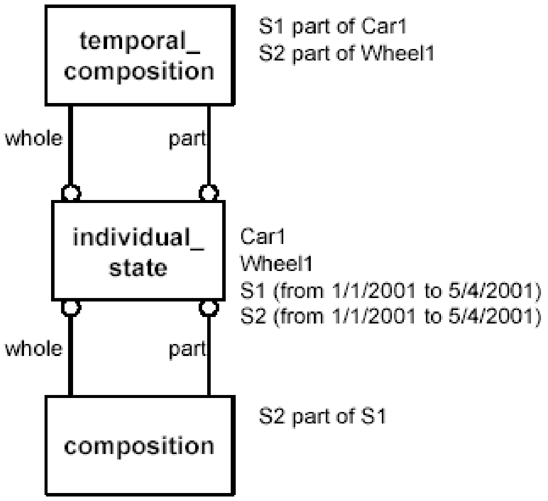


Figure E.8 — Composition using states

Here the individual_states S1 and S2 are modelled explicitly. S1 is shown as being a temporal part of Car1, S2 is shown as being a temporal part of Wheel1, and S2 is shown as being a part of S1.

E.3.3 Pattern 3: coincident individuals

Figure E.9 shows how a particular pump, Pump 1234, is installed to perform a particular duty, TAG P101, for a period of time between 3/1/2000 and 5/8/2001. At the end of this period, the pump is removed and replaced by another one performing the same duty.

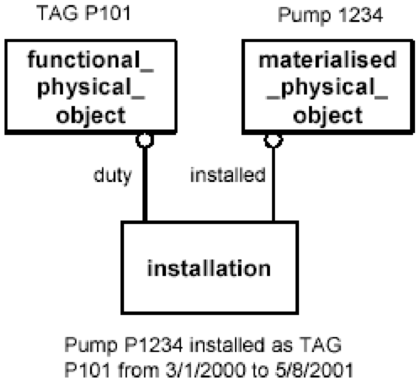

TAG P101
Pump 1234
functional_ physical_ object
materialised _physical_ object
duty
installed
installation
Pump P1234 installed as TAG P101 from 3/1/2000 to 5/8/2001

Figure E.9 — Coincident individuals

Figure E.10 shows this example using a space-time map. Here it is possible to see that the duty represented by TAG P101, and Pump 1234 are coincident for the period of the installation, i.e. the state S1 of Pump 1234 that is installed as TAG P101 is in fact also a state of TAG P101. Indeed, the TAG P101 consists of those states of the pumps that are installed in this place.

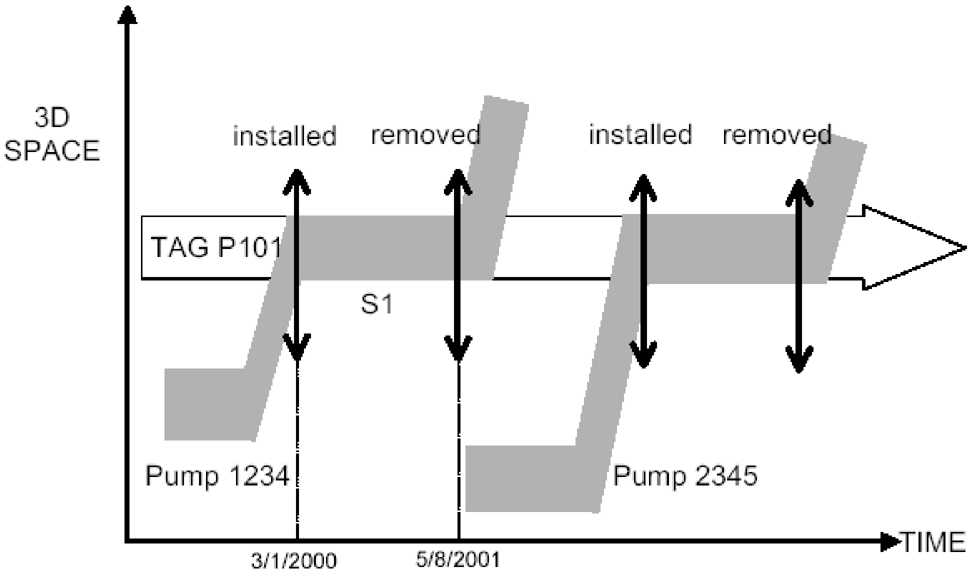


Figure E.10 — Space-time map for coincident individuals.

When this is modelled explicitly, rather than as an association, the model in Figure E.11 results. Here S1 is shown as being a temporal part of both TAG P101 and Pump 1234.

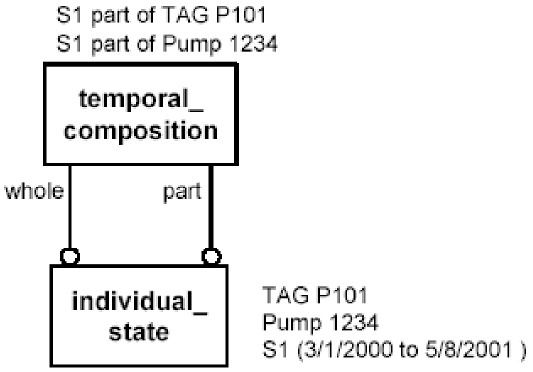


Figure E.11 — Coincident individuals using states

E.3.4 Pattern 4: a relationship between two classes

Figure E.12 shows a specialization association, and an example is given that says that a centrifugal pump is a specialization of pump. The association automatically allows the specialization to have a start and end date, but in fact classes are timeless, and there is no time when centrifugal pump is not a specialization of pump.

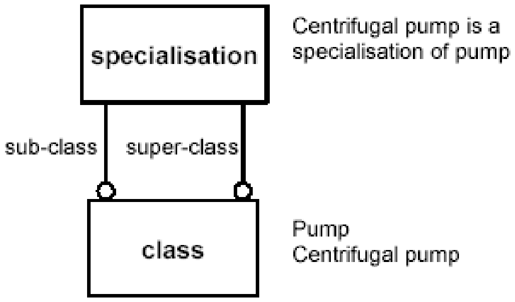


Figure E.12 — A relationship between two classes

In this case there is no space-time map to draw, since all the objects sit outside of time. However, we can examine what the specialization association means, and in this case we can determine that the specialization association indicates that each member of the sub-class is also a member of the superclass. We therefore identify specialization as a subtype of relationship as illustrated in Figure E.13.

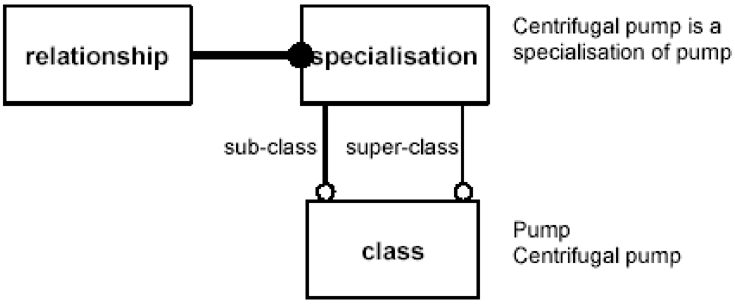


Figure E.13 — Analysis of a relationship between two classes

E.3.5 Pattern 5: a class of relationship between two classes

Figure E.14shows an association between two classes that exhibits a different pattern. The association is composition_according_to_class, which is used to say that a member of the whole class, has a member of the part class as a part. The example given is that a centrifugal pump has an impeller as a part. Again, although an association can have a start and end date, none applies to this, as the statement is always intended to be true.

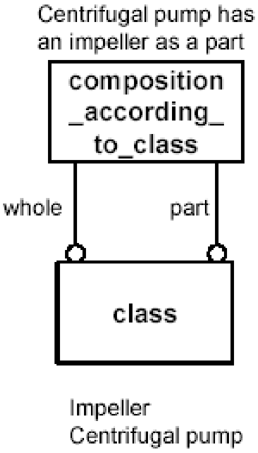

Centrifugal pump has an impeller as a part
composition_according_to_class
whole
part
class
Impeller
Centrifugal pump

Figure E.14 — A class of relationship

Once again there is no space-time map, but analysis reveals a different pattern from the previous example. The association represents a rule again. In this case it is that each member of the whole class may have a member of the part class as a part. So each centrifugal pump may have an impeller as a part. Also in this case there will be particular relationships between particular impellers and particular centrifugal pumps that are instances of this rule. Hence, this association represents a class_of_relationship.

This is illustrated in Figure E.15. Here the composition relationship and the classification relationships are added that show how the rule applies to particular instances. An impeller state, Imp 3456 S1 is classified as an impeller; a pump state, Pump 1234 S2 is classified as a pump; a composition relation #1 shows that Imp 3456 S1 is a part of Pump 1234 S2; finally a classification of relationship shows that #1 is a case of an impeller being a part of a centrifugal pump.

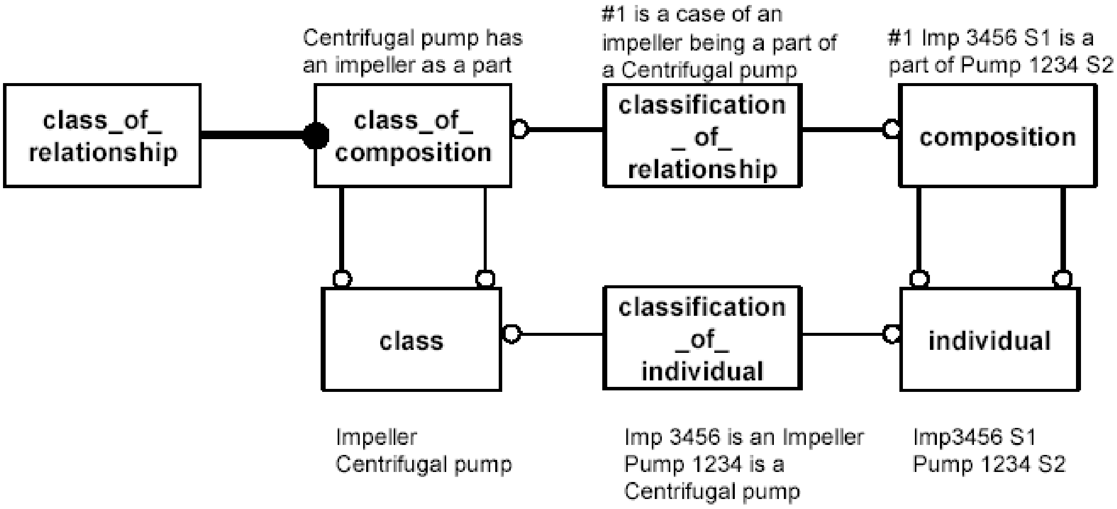

Centrifugal pump has an impeller as a part
#1 is a case of an impeller being a part of a Centrifugal pump
#1 Imp 3456 S1 is a part of Pump 1234 S2
class_of_ relationship
class_of_ composition
classification _ of_ relationship
composition
class
classification _of_ individual
individual
Impeller
Centrifugal pump
Imp 3456 is an Impeller
Pump 1234 is a Centrifugal pump
Imp3456 S1
Pump 1234 S2

Figure E.15 — Analysis of a class of relationship

E.4 Conclusions

Associations have been a powerful way to help in managing history, as opposed to just the current state of a domain. However, the analysis presented in this paper shows that considerable detail is hidden by the association construct. This paper shows that undertaking spatio-temporal analysis of associations allows a more precise model to be developed which makes this hidden detail more explicit. In particular five patterns of how associations can map into spatio-temporal terms have been identified and presented.

Bibliography

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

[2] ISO/TR 9007:1987, *Information processing systems — Concepts and terminology for the conceptual schema and the information base*.

[3] ACZEL, Peter. *Non-Well-Founded Sets*, Center for the Study of Language and Information, Stanford, California, 1988, ISBN 0937073229.

[4] ITÔ, K. (editor). *Encyclopedic Dictionary of Mathematics*, Mathematical Society of Japan, Edition 2, Cambridge, Massachusetts, MIT Press, 1993, ISBN 0262590204.

[5] WEST, Matthew; FOWLER, Julian. *Developing High Quality Data Models*. Version 3.0. EPISTLE, 1996-08-27 [cited 2001-03-11]. Available from the World Wide Web: <http://www.stepcom.ncl.ac.uk/epistle/data/mdlgdocs.htm>.

[6] WEST, Matthew. *Some Notes on the Nature of Things*. ISO TC184/SC4/WG10 N307, 2000-06-09 [cited 2001-03-11]. Available from the World Wide Web: <http://www.nist.gov/sc4/wg_qc/wg10/current/n307/wg10n307.htm>.

Index

ICS 25.040.40; 75.020

Price based on 225 pages