INTERNATIONAL STANDARD

ISO 19440

First edition
2007-12-01

Enterprise integration — Constructs for enterprise modelling

Entreprise intégrée — Constructions pour la modélisation d'entreprise

Reference number
ISO 19440:2007(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.

COPYRIGHT PROTECTED DOCUMENT

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

Page

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 19440 was prepared by the European Committee for Standardization (CEN) Technical Committee CEN/TC 310, *Advanced manufacturing technologies*, in collaboration with Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

Introduction

0.1 Background

This International Standard defines the generic concepts that are required to enable the creation of enterprise models for industrial businesses and to provide support for the use of frameworks by industrial enterprises. This International Standard builds upon ISO 19439 by defining and detailing a set of conformant user-oriented modelling language constructs, which provide common semantics and enable the unification of models developed by different stakeholders in the various phases of model development. Such models are aimed at model-based support of operational decision-making and can be employed for model-based operation monitoring and control.

The modelling language constructs defined in this International Standard can be specialized or organized or both into structures for specific purposes, for example for an industry sector or for a particular kind of enterprise concern such as maintenance. In turn, such structures and generic modelling language constructs can be used for developing particular models for a particular enterprise. Annex B contains further background, the rationale and benefit statements for this International Standard.

The general requirements that determine the characteristics of the core constructs necessary for computer-supported modelling of enterprises are

— the provision of an explicit model of Business Processes, with their dynamics, functions, information, resources, organization and responsibilities,

— sufficient detailing and qualification of enterprise components to allow the creation of a model for a specific enterprise,

— support for management of change, and

— end-user-oriented representation to enable operational use.

An illustrative example shown in Annex E demonstrates the use of the modelling language constructs.

Annex B provides a rationale for construct-based enterprise modelling and sets out the background to this International Standard and the framework for enterprise models on which it is based (see ISO 19349). The three dimensions of this framework are described in 0.2, 0.3 and 0.4 below.

Contributions to this International Standard have been received from members of the IFAC/IFIP Task Force on Enterprise Integration, the CIMOSA consortium and the European ATHENA research project.

NOTE Figures C.1 to C.6, D.1 to D.3 and D.5 to D.12 are computer-generated. Figures D.4 and E.1 to E.5 are line drawings.

0.2 Dimension of enterprise model views

ISO 19439 and ISO 15704 use enterprise model views (often shortened to “model views”) to provide a selective perception of an Enterprise that emphasizes some particular aspect of the matter under consideration and disregards others. Specifically, they identify four enterprise model views (Function, Information, Resource, Organization) that are to be addressed in a framework, architecture or methodology to allow the modelling of the major aspects of an enterprise. Additionally, as stated in ISO 15704:2000, A.3.1.5.3.2, “other ... views may be defined if needed ... and supported by the engineering tools”, e.g. economic views, decision views, purpose views and implementation views. In this case, the constructs defined in this International Standard can be augmented by additional attributes to support these other views, or

relevant new constructs might have to be defined. Therefore, the specifications of modelling language constructs have to accommodate their intended usage and representation in one or several particular model views. Automated tools are required to ensure consistency of construct instances that can appear in more than one view.

0.3 Dimension of enterprise model phase

The life cycle of models and model components is addressed by the dimension of the enterprise model phase in ISO 19439. This dimension is concerned with the development and evolution of the model of the domain to be modelled, starting from the identification of the enterprise domain and progressing to a processable model and the decommissioning thereof. Therefore, the specifications of modelling language constructs have to accommodate their intended usage and representation in a particular model phase. Attributes of modelling language constructs need to be adaptable and selectable for the different model phases according to the envisioned needs.

0.4 Dimension of genericity

Relative to the dimension of genericity defined in ISO 19439, constructs reside at the generic level and can be used at the partial and particular levels. At the partial level some attribute values can remain undefined for partial instances (e.g. inputs/outputs for Events for Domains and inputs/outputs for Business Processes). Such missing entries have then to be completed at the particular level.

Enterprise integration — Constructs for enterprise modelling

1 Scope

This International Standard specifies the characteristics of the core constructs necessary for computer-supported modelling of enterprises conforming to ISO 19439.

This International Standard focuses on, but is not restricted to, the computer integration of the information aspects of manufacturing, including the management and control technology and the required human tasks. It does not specify how these core constructs for model-based operations are to be implemented and, in particular, it does not include the control language needed to specify and execute (internal) activity behaviour, nor the mapping between functional operations and capabilities.

NOTE Computer-supported modelling of enterprises can form a precursor to computer integration or human-system intermediation.

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/IEC 14977, *Information technology — Syntactic metalanguage — Extended BNF*[1)]

ISO 19439:2006, *Entreprise integration — Framework for enterprise modelling*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Terms defined in ISO/IEC 15288:2002, ISO 15531-1:2004, ISO 15704:2000 and ISO 19439:2006 and used in this International Standard are repeated below for convenience.

NOTE The names of terms representing constructs are capitalized throughout this International Standard to aid the reader in distinguishing them from general usages of the same term, specifically in order to distinguish the constructs Capability, Domain, Enterprise Activity, Event and Resource from general usage of capability, domain (or enterprise domain), enterprise activity, event and resource. These constructs are defined in 3.1 and specified in Clause 6.

3.1.1
aggregation
process of, or result of, combining modelling language constructs and other model components into a whole entity

1) ISO/IEC 14977 is a freely available International Standard that can be downloaded free of charge from http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm.

NOTE 1 Modelling language constructs and other model components can be aggregated into more than one entity.

NOTE 2 Both Part_of and Consists_of attributes are used in the aggregation relationships described in Clause 5.

[ISO 19439:2006]

3.1.2
attribute
piece of information stating a property of an entity

[ISO 15704:2000]

3.1.3
behavioural rule
description of the logical sequencing relationships of constituent activities used in the specification of Business Process behaviour

3.1.4
Business Process
⟨enterprise modelling⟩ construct that represents a partially ordered set of Business Processes or Enterprise Activities, or both, that can be executed to realize one or more given objectives of an enterprise or a part of an enterprise to achieve some desired end-result

3.1.5
capability
⟨general⟩ quality of being able to perform a given activity

[ISO 15531-1:2004]

3.1.6
Capability
⟨enterprise modelling⟩ construct that represents the collection of capability characteristics (expressed as capability attributes) of either a Resource (its provided Capability) or an Enterprise Activity (its required Capability)

NOTE Capabilities can be aggregated.

3.1.7
class
abstraction representing and encapsulating properties, relationships and behaviour, which distinguish a collection of similar phenomena

NOTE Class is used in a very general sense without any connotation for implementation or for use with a specific methodology.

3.1.8
classification
process of arranging abstractions into a structure, organized according to their distinguishing properties, relationships and behaviour

3.1.9
component
⟨general⟩ entity that is part of, or capable of becoming part of, a larger whole

NOTE Adapted from ISO 19439:2006.

3.1.10
component
⟨system⟩ entity, with discrete structure within a system, that interacts with other components of the system, thereby contributing to the system properties and characteristics

NOTE Adapted from ISO/IEC 15288:2002.

3.1.11
concept definition
enterprise model phase that defines the business concepts of an enterprise domain to be employed in realizing its business objectives and its operation, including the necessary enterprise domain inputs and outputs

[ISO 19439:2006]

3.1.12
constraint
restriction or limitation or condition placed upon a system that originates from inside or outside the system under consideration

[ISO 19439:2006]

3.1.13
construct-based modelling language
set of constructs and rules for valid groupings, which define the syntax of the modelling language

3.1.14
construct label
literal string defined for each construct template, denoting the kind of construct

3.1.15
construct template
common structure that allows the identification and description of particular modelling language constructs and the assignment of their properties

3.1.16
declarative rule
set of objectives and constraints combined with a non-computational set of conditions

NOTE Declarative rules can be imposed within Domains on Business Processes.

3.1.17
Decision Centre
⟨enterprise modelling⟩ construct that represents a set of decision-making activities that are characterized by having the same time horizon and planning period and belonging to the same kind of functional category

NOTE The terminology used to describe aspects of Decision Centre is found in ISO 15704:2000/Amd.1:2005, Annex C, which defines [time] horizon as “the part of the future taken into account by a decision, i.e. the horizon is six months when a decision is taken on a time interval of six months” and [planning] period as “the time that passes between a decision and when this decision shall be re-evaluated”.

3.1.18
decommission definition
enterprise model phase that defines the final state of a decommissioned operational system, all its components for a particular enterprise domain and the processes employed to conduct the decommissioning, so enabling reuse or disposition of those components

[ISO 19439:2006]

3.1.19
derivation
⟨enterprise modelling⟩ process of elaborating enterprise models at successive enterprise model phases from the models established at preceding phases, reusing the available contents and extending them according to the needs expressed for the particular model phase

3.1.20
design specification
enterprise model phase that specifies the Business Processes together with Capabilities and rules that are to be performed to achieve the requirements

NOTE Adapted from ISO 19439:2006.

3.1.21
Domain
⟨enterprise modelling⟩ construct that represents the portion of an enterprise to be modelled providing for identification of the relevant information

3.1.22
domain identification
enterprise model phase that identifies the enterprise domain to be modelled with respect to its business objectives, the enterprise domain inputs and outputs and their respective origins and destinations

NOTE Adapted from ISO 19439:2006.

3.1.23
domain operation
enterprise model phase that encompasses the operational use of the domain model

NOTE Adapted from ISO 19439:2006.

3.1.24
enterprise activity
⟨general⟩ all, or part, of the lowest level of process functionality required by user objectives that consists of functional operations performed in the enterprise that consume inputs and allocate time and resources to produce outputs

NOTE Adapted from ISO 19439:2006.

3.1.25
Enterprise Activity
⟨enterprise modelling⟩ construct that represents a certain part of the lowest level of enterprise functionality required by user objectives and identifies the inputs needed for its execution and the outputs created as a result

3.1.26
enterprise domain
part of the enterprise considered relevant to a given set of business objectives and constraints for which an enterprise model is to be created

NOTE In this International Standard, “enterprise domain” is abbreviated to “domain” whenever it is used as a qualifier in such terms as “domain identification phase” and “domain model”. Other usages of “domain” have the normal dictionary meaning.

3.1.27
enterprise model
abstraction that represents enterprise entities, their interrelationships, decomposition and detailing, to the extent necessary to convey what the enterprise intends to accomplish and how it operates

NOTE Adapted from ISO 19439:2006.

3.1.28
enterprise model phase
life cycle phase of an enterprise model

[ISO 19439:2006]

3.1.29
enterprise model view
model view
selective perception or representation of an enterprise model, which emphasizes some particular aspect and disregards others

[ISO 19439:2006]

3.1.30
Enterprise Object
⟨enterprise modelling⟩ construct that represents a piece of information in the enterprise and that describes a generalized or a real or an abstract entity which can be conceptualized as being a whole

NOTE 1 The usage of Enterprise Object is restricted to those situations where only the information aspects of the entity under consideration are relevant.

NOTE 2 All other constructs in this International Standard represent entities that have specific semantics requiring particular attributes and additional descriptions.

NOTE 3 Adapted from ISO 19439:2006.

3.1.31
Enterprise Object View
Object View
⟨enterprise modelling⟩ construct that represents a collection of attributes of an Enterprise Object for some particular purpose

NOTE The collection is defined by a selection of attributes and possibly constraints on those attributes.

3.1.32
entity
any concrete or abstract thing in the domain under consideration

[ISO 19439:2006]

3.1.33
Event
⟨enterprise modelling⟩ construct that represents a solicited or unsolicited fact indicating a state change in the enterprise or its environment

NOTE An event can be associated with an Object View containing information related to the Event.

3.1.34
function view
enterprise model view that enables the representation and modification of the processes of the enterprise, their functionalities, behaviour, inputs and outputs

[ISO 19439:2006]

3.1.35
Functional Entity
⟨enterprise modelling⟩ construct that is a specialization of the Resource construct, which represents an aggregation of Resources and Operational Roles able to perform, completely on its own, a (class of) functional operation(s) required by an Enterprise Activity and to communicate with the related control system

NOTE A characteristic of a functional entity is its ability to receive, process, store and send information.

3.1.36
functional category
grouping of entities for expression of a common purpose or capability

3.1.37
functional operation
basic unit of work and lowest level of granularity in the function view

NOTE Functional operations are identified in the design specification phase following from a decomposition of the required capabilities of an Enterprise Activity (the task) into subtasks that can then be matched to the capabilities provided by the assigned Functional Entities.

3.1.38
generalization
specific concept modified for a more general extent, use or purpose, or act of removing or modifying detail from a specific concept to produce a generalization thereof

NOTE 1 Generalization is the inverse of specialization.

NOTE 2 Adapted from ISO 19439:2006.

3.1.39
generic level
collection of generic modelling language constructs for expressing descriptions that can be used to generate models at the partial and particular levels

[ISO 19439:2006]

3.1.40
implementation description
enterprise model phase that describes the final set of processes, resources and rules implemented to achieve the desired operational performance for execution of Business Processes and Enterprise Activities specified in the design specification phase

[ISO 19439:2006]

3.1.41
information technology component
component that is required to undertake one or several of the collection, processing, distribution, storage or verification of data for Enterprise Activities in the enterprise

NOTE Adapted from ISO 19439:2006.

3.1.42
information view
enterprise model view that enables the representation and modification of the enterprise information as identified in the function view

NOTE 1 It is organized as a structure containing enterprise objects that represent the information-related entities of the enterprise (material and information).

NOTE 2 Adapted from ISO 19439:2006.

3.1.43
instantiation
creation of instances of modelling language constructs or partial models and the possible assignment of values to some or all attributes

NOTE A fully instantiated modelling language construct or model is one for which values have been assigned to all attributes.

[ISO 19439:2006]

3.1.44
integrate
ensure the interaction between all enterprise entities that is necessary to achieve a given purpose in a given constrained environment

NOTE Integration is the result of this method.

3.1.45
integrity rule
statement in the requirements definition phase concerning restrictions on information to ensure conformity to real-world reality.

NOTE Integrity rules are used to define these restrictions in terms of constraints on attributes of Enterprise Objects.

3.1.46
life cycle
set of distinguishable phases and steps within phases that an entity goes through from its creation until it ceases to exist

[ISO 19439:2006]

3.1.47
life cycle phase
stage of development in the life cycle of an entity

NOTE Adapted from ISO 19439:2006.

3.1.48
manufacturing technology component
component that is required to undertake one or several of the control, transformation, transport, storage or verification of raw materials, parts, (sub-)assemblies and end products

NOTE Adapted from ISO 19439:2006.

3.1.49
model
abstract description of reality in any form (including mathematical, physical, symbolic, graphical, or descriptive) that presents certain aspects of that reality

NOTE Adapted from ISO 19439:2006.

3.1.50
modelling language construct
construct
textual or graphical part of a modelling language devised to represent, in an orderly way, the diverse information on common properties and elements of a collection of enterprise entities

NOTE Adapted from ISO 19439:2006.

3.1.51
objective
statement of preference about possible and achievable future situations that influences the choices within some behaviour

[ISO 19439:2006]

3.1.52
occurrence
⟨enterprise modelling⟩ single, actual realization of a modelling language construct that represents a particular entity in the real world at the time the model is processed

3.1.53
Operational Role
⟨enterprise modelling⟩ construct that represents the relevant human skills and responsibilities required to perform those operational tasks that are assigned to the particular Operational Role

3.1.54
Order
⟨enterprise modelling⟩ construct that is a specialization of the Enterprise Object construct, which represents the information for planning and control of Business Processes in an enterprise

NOTE An Order can be represented by an Object View that is associated with an Event.

3.1.55
Organizational Role
⟨enterprise modelling⟩ construct that represents, within a given hierarchical structure of an enterprise, the organizationally relevant human skills and responsibilities required to perform those organizational tasks that are assigned to the particular Organizational Role

3.1.56
Organizational Unit
⟨enterprise modelling⟩ construct that represents an entity of the organizational structure of an enterprise, which is described by attributes of the organization and references to both lower- and higher-level organizational entities

NOTE Examples of Organizational Unit could be department or division.

3.1.57
organization view
enterprise model view that enables the representation and modification of the organizational and decisional structure of the enterprise and the responsibilities and authorities of the persons and organization units within the enterprise

NOTE Adapted from ISO 19439:2006.

3.1.58
partial level
collection of partial models

[ISO 19439:2006]

3.1.59
partial model
model used as a reference model in a specific type of industry segment or industrial activity

NOTE 1 A partial model is comprised of modelling language constructs or other partial models. Partial models also enable a modeller to reuse already existing models built for other domains.

NOTE 2 Adapted from ISO 19439:2006.

3.1.60
particular level
level at which a model is described for a particular, specific enterprise domain

[ISO 19439:2006]

3.1.61
particularization
process of specialization and instantiation by which more specific model components can be derived from more generic ones

[ISO 19439:2006]

3.1.62
performance indicator
a metric or measure by which the achievement of an objective can be assessed

3.1.63
Person Profile
⟨enterprise modelling⟩ construct that represents a set of personal skills and responsibilities that are required by an Organizational Unit or an Enterprise Activity, or both, and that are provided by a person

NOTE Person Profiles can be assigned to more than one person and conversely a person can fulfil more than one Person Profile for more than one Organizational Unit or Enterprise Activity.

3.1.64
processable model
model with specified syntax and semantics, which can be processed by a computer (for analysis, simulation or execution)

3.1.65
Product
⟨enterprise modelling⟩ construct that is a specialization of the Enterprise Object construct, which represents the desired output or by-product of the Business Processes of an enterprise

3.1.66
relationship
association between two or more entities that is significant for some intended purpose

3.1.67
requirements definition
enterprise model phase that defines the enterprise operations needed to achieve enterprise objectives and the conditions necessary to enable those operations, both being without reference to implementation options or implementation decisions

NOTE Adapted from ISO 19439:2006.

3.1.68
resource
⟨general⟩ enterprise entity that provides some or all of the capabilities required to execute an enterprise activity

NOTE In this International Standard, resource is used in the system theory sense of entities that provide capabilities required by the system and are an essential part of the system itself. The resource description includes the identification and description of consumables (such as energy, air, coolant) that are required to be present in sufficient quantities to operate the resource. In contrast, material is reserved for process inputs that are required by the various processes, such as raw materials, parts and assemblies. These inputs are identified in the function view, described in the information view, and have the associated management responsibilities identified in the organization view.

[ISO 19439:2006]

3.1.69
Resource
⟨enterprise modelling⟩ construct that is a specialization of the Enterprise Object construct, which represents the provided capabilities available to execute an Enterprise Activity

NOTE The Resource construct does not include human resources.

3.1.70
resource view
enterprise model view that enables the representation and modification of enterprise resources and persons

NOTE 1 The resource view is organized as a structure containing Enterprise Objects representing the set of resources and set of persons required to execute enterprise operations.

NOTE 2 Adapted from ISO 19439:2006.

3.1.71
specialization
general concept modified for a more limited extent, specific use or purpose, or act of adding to, or modifying details of, a general concept to produce a specialization thereof

NOTE 1 Specialization is the inverse of generalization.

NOTE 2 When referring to a class, specialization involves the construction of subclasses within a class for a particular purpose, where the members of each subclass have one or more characteristics (attributes, relationships, behaviour or semantics) in common which are not shared by all other members of the class. When referring to a model, it refers to the progression from generic concepts to partial models and particular models.

NOTE 3 Adapted from ISO 19439:2006.

3.1.72
system
collection of real-world items organized for a given purpose

[ISO 19439:2006]

3.1.73
universe of discourse
collection of entities that ever have been, are, or always will be, in a selected portion of the real world or postulated world of interest that is being described by the models

3.2 Abbreviated terms

ATHENA Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications

CIM Computer Integrated Manufacturing

GRAI Graphes à Résultats et Activités Interreliés (Graphs of Interrelated Results and Activities)

IFAC International Federation of Automatic Control

IFIP International Federation for Information Processing

ODP Open Distributed Processing

POP* Process – Organization – Product – Others

PSL Process Specification Language

XML eXtensible Markup Language

4 Common characteristics of modelling language constructs

Modelling languages that are based on a set of common generic modelling language constructs simplify the creation of enterprise models, increase modelling efficiency and improve model understanding and interoperability across organizations and even across industries. Such constructs should be capable of representing the many different entities that exist in enterprises.

NOTE 1 The constructs defined in this International Standard are set out in Clause 6. A construct is an element of a modelling language, the syntax and semantics of which are defined as precisely as possible for the purposes of compliance testing as described in Clause 7. However, in its present stage of development this International Standard does not propose a formal notation for these purposes.

Adaptation of these constructs to different stakeholder languages appropriate for their purposes might require different forms of representation for some enterprise model phases, but such forms should maintain to the extent possible the underlying semantics of the constructs. In addition, the set of generic modelling language constructs may be augmented by more specialized construct classes to increase the modelling efficiency or to address requirements of any additional modelling views.

This International Standard is concerned with business process-oriented enterprise modelling. Therefore, the enterprise entities to be represented shall be Business Processes with their dynamics (control flow/process behaviour), functionalities (activities, inputs/outputs, control, resources) and organizational aspects. These entities shall be represented at the different model phases and described at the appropriate level of detail. In addition, the representation of aggregations shall be supported not only for the Business Process/Enterprise Activity relation, but also for the enterprise entities that are affected by those Business Processes. Function-, information-, resource- and organization-model views shall be used to provide the foundation for user-oriented modelling. Those model views shall allow the identification of relevant object hierarchies and relationships between the different classes and subclasses.

NOTE 2 EXPRESS [1], PSL [9] and Petri nets [8] are single-view languages, while the Enterprise Language of ODP [4], the ATHENA POP* [10] and this International Standard support multi-view enterprise modelling. Petri nets target general discrete event systems modelling, including manufacturing systems. PSL is a neutral language for specifying manufacturing process information. EXPRESS is an abstract language (with syntax and semantics) for modelling a population of entities and their data attributes. Petri nets, PSL and EXPRESS are formal languages that are computer executable, while ODP's Enterprise Language and the constructs of this International Standard are neither. ODP [4], POP* [10] and this International Standard do not have the same viewpoints (known in this International Standard as model views) and coverage. POP* and this International Standard adopt a more process-oriented modelling approach aimed at producing an operational process monitoring and control model. In contrast, ODP has a broader coverage with its five viewpoints (Enterprise, Information, Computation, Engineering and Technology). The ODP Enterprise viewpoint is most relevant to this International Standard, focusing as it does on enterprise level concepts such as the purpose, scope and policies of the system.

Modelling of the enterprise shall be supported at all enterprise model phases by modelling language constructs as follows:

— at the domain identification phase, to identify the contents of the domain and its inputs and outputs, in such a way that the domain identification model can be used as the starting point for a derivation of a consistent concept definition model;

— at the concept definition phase, to define underlying missions, strategies, etc., and in such a way that the concept definition model can be used as the starting point for a derivation of a consistent requirement definition model;

— at the requirements definition phase, to describe the Business Processes of the Domain from a business viewpoint, in such a way that the requirements definition model can be demonstrated to be sufficient for use as the starting point for the derivation of a consistent design specification model. Therefore the requirements definition model shall also be capable of being processed;

— at the design specification phase, to specify the Business Processes with all their components from both a business and ICT viewpoint, in such a way that the design specification model can be demonstrated to be sufficient for use as the starting point for a consistent derivation of an implementation description

model: therefore the design specification model shall also be capable of being processed; at the design specification phase, constructs shall be derived from those of the requirements definition phase by enriching them with attributes that reflect all resources and general ICT interfaces;

— at the implementation description phase, to describe the Business Processes with all their components as they are implemented in the actual system from both a business and ICT viewpoint, in such a way that the implementation model can be demonstrated to be sufficient for use in the operation of the enterprise for decision support and Business Process monitoring and control: therefore the implementation model shall also be capable of being processed; at the implementation description phase, constructs shall reflect the hardware/software platform or technology chosen to be consistent with design specification constructs;

— at the domain operation phase, to use the released model for operational purposes such as decision support, monitoring and control;

— at the decommission definition phase, to identify future use of the Business Process components, in such a way that the decommission definition model can be used, as appropriate, as the starting point for reuse at any model phase.

However, this does not mean there will be a different set of modelling language constructs required for each model phase. Constructs and construct instances are carried forward to subsequent model phases by providing an enhanced attribute list to capture additional needed information for representing and describing the enterprise entities at those model phases or by mapping onto different construct representations. Clause E.2 summarizes the identification and usage of constructs in each model phase.

NOTE 3 The applicability of the modelling language constructs is demonstrated in Annex D, using ODP and the ATHENA work on POP* as examples.

5 Representations, relationships, roles and complementary concepts

5.1 Range of representation

Modelling language constructs shall be represented in a form that is both understandable by humans and readable by machines[2]. This means there is the need for both a visual (graphical, iconic) representation of structures [3] for use by model builders and business users, and a structured information-capturing representation for which this International Standard uses templates.

Representation shall be performed in the following four ways:

— representation by a common structure of the constructs of this International Standard (see 5.2);

— representation of the relationships between these modelling language constructs (see Annex C[4]);

— representation by templates to organize attributes into a common format (see Clause 6);

— representation of dynamic behaviour by behavioural rules (see 6.3.5 and Annex A).

2) Machine readability is not the subject of this International Standard, but has to be provided by the modeling tool.

3) Representation of modelling language constructs by graphics is not part of this International Standard.

4) Annex C contains an illustrative UML metamodel as an aid to the reader.

5.2 Common structure and template for modelling language constructs

To describe a particular modelling language construct, this International Standard uses a common structure that provides a consistent minimal assignment of attributes to each construct. This common structure shall have two constituents:

— a textual description consisting of brief text defining the modelling language construct in terms of a purpose, its description and its intended usage, and

— a construct template, which organizes and defines the attributes for this modelling language construct.

The template is described in an informal manner but using a common pro forma. In turn, the construct template shall have the form described below.

a) Header part having the same attributes for each modelling language construct and containing attributes relating to the identity of a construct instance and to its context. It shall be structured as follows:

   1) construct label (a literal string denoting the kind of construct);

   2) identifier (a literal string that is unique for each occurrence of the modelling language construct within the model);

   3) name (the name of the instance of the modelling language construct);

   4) authority for model design (i.e. the identifier of the Organizational Role or Organizational Unit responsible for the design of this construct): for all constructs and for all attributes concerned with authorities or responsibilities, the identifier or name of the Organizational Unit may be omitted in the Concept Identification and Requirements Definition modelling phases.

NOTE These are the only phases in which NIL is an option for Organizational Unit. NIL is also an option for a Domain's Organizational Role, but not for any other construct.

b) Body part containing the particular attributes that are specific to each construct and whose description is derived from the particular modelling language construct definition. Body parts shall then be structured in two further partitions as follows:

   1) descriptives, containing descriptive attributes that comprise

      — construct-identifying description in textual form,

      — construct attributes that are predefined,

      — additional construct attributes that may be defined by the user to meet particular needs, such as those required by additional model views, and

      — attribute qualifications – statements that are made about whether attribute values are mandatory or optional, when they are applicable, etc.

   2) relationships (further described in 5.4), containing relationship attributes that can include

      — Operational relationships, that are the responsibility and authority for model operation (i.e. the identifier of the Organizational Role and Organizational Unit responsible for the operational usage of this construct or authorized to change its usage),

      — Specialization_of relationships, representing relationships between a specialization and its generalization,

      — Part_of relationships, representing relationships between this construct instance and the whole aggregated from such instances,

— Consists_of relationships, representing relationships between this construct instance and its constituent parts[5], and

— Association relationships for other forms of relationship, either predefined or user-defined (such as provision or usage of a Capability instance by Resource or Enterprise Activity instances).

Both descriptives and relationships contain general attributes relevant for most or all enterprise model phases. Constraints on the relevance of attributes are indicated in the construct templates and general attributes are applicable to all enterprise model phases unless otherwise indicated. Both descriptives and relationships may also contain specific attributes and constraints that are relevant only for selected enterprise model phases. Additionally, both descriptives and relationships may be grouped as sets of attributes having some common characteristics, e.g. inputs/outputs or relationships to organizational concerns.

Version control of modelling language constructs is not supported by a specific attribute since it is seen to be a tool-provided responsibility[6].

NOTE 1 In the models of Annex C, Part_of and Consists_of relationships are shown explicitly, rather than as UML aggregations. UML associations are used to show Operational and Association relationships, and UML generalizations are used to show Specialization_of relationships.

NOTE 2 It is the responsibility of the tool-supporting software development to ensure consistency of Part_of and Consists_of relationships where both are used, and of Constraints and Integrity descriptives.

5.3 Representation of attributes

Each attribute shall be defined by its name and predefined data type (numbers, literals, strings, etc.).

NOTE 1 A data type can take on a range of permissible or actual values, such as a single identifier (id = 4711), value range (diameter 10 mm $\pm$ 0,05 mm) or value list (colour is yellow, red or green).

NOTE 2 Constructs can use XML schema [13],[14],[15] or EXPRESS [1] as a means for representing attributes and the related definition of data types, expressions and statements.

5.4 Representation of relationships

The purpose of a relationship is to model time-varying and other associations between run-time instances of modelling language constructs. Reflexive relationships shall be permitted, i.e. a relationship can be created between an instance and itself.

EXAMPLE 1 Relationships between entities might be those between a part and the machine which is used to produce it, or those between a worker and the order to which he or she has been assigned, as in: Produced_by (part, machine), Works_at (worker, order). Such relationships are represented through assignments as inputs and outputs of Enterprise Activities.

EXAMPLE 2 Illustrative relationships between constructs are those between a Domain and its Business Processes or those between a Business Process and its sub-processes and activities.

The concept of a user-defined relationship is dependent on the concept of Enterprise Object described in 6.6 and on the model entities derived from that Enterprise Object. A user-defined relationship is used to describe domain-dependent relationships as perceived and defined by the users, and hence to describe the way in which they group structuring and behavioural constructs in terms of their utilization within the enterprise. User-defined relationships can be expressed by Related_to lists in Enterprise Objects, Products, Orders and Object Views on Enterprise Object instances, or by additional user-defined attributes.

---

5) It is accepted that Part_of and Consists_of are complementary relationships; depending on the purposes of the modeller, either or both may be used.

6) Version control has a significant impact on design and operation for enactable models.

5.5 Specializations

Depending on the purpose of modelling, it may be helpful and necessary for the modeller to define specializations of the constructs defined in this International Standard. This can be achieved by

— selecting the most appropriate construct (and corresponding template) to specialize and adapting the header information accordingly,

— adding a new “Specialization_of” relationship attribute at the appropriate model phases(s),

— assigning appropriate values for the “Specialization_of” attribute and listing the construct (or, exceptionally constructs) of which it is a specialization,

— assigning or constraining values for existing attributes as appropriate, and

— adding additional attributes as required and assigning model-unique names for these.

Each construct can be further specialized into constructs of the same kind; for example an Enterprise Object can be specialized into particular kinds of Enterprise Object, Resource into further kinds of Resource, etc. A relationship “Specialization_of” is introduced to hold these specialization relationships.

Additionally, this International Standard predefines several specializations. Three constructs are specializations of the Enterprise Object construct (Product, Order, Resource), and a further construct (Functional Entity) is a specialization of Resource. This specialization can occur because the attributes and relationships of Enterprise Object exist whenever the object instance exists, even if some of those attributes and relationships are not shown as applicable to a particular modelling phase.

NOTE The concept of specialization used in this International Standard is less constrained than the specialization concept found in the domain of programming languages and object-oriented modelling.

5.6 Roles

The Oxford English Dictionary [16] defines a role as a “typical or characteristic function performed by someone or something”. The same entity may have different roles in different contexts (which may overlap in time and place) and at different times over its life cycle. This role variation is especially true for the following examples of entities:

— for human resources, an employee can be in turn a subordinate in the structure of a given organization, a manager of other employees in the same structure and an external part-time consultant in the structure of another organization; this person may then provide different sets of capabilities when involved in his or her different roles;

— a machine has a different role when it is used as an operational resource providing machining functionalities during its normal utilization as opposed to when it is under maintenance or becoming a product as the result of its manufacturing process;

— for IT resources, a computer application can in turn be a client of another application and a server of a different application.

The modelling language constructs defined in this International Standard provide for the representation of roles as follows:

a) the role of humans when supporting an Organizational Unit shall be represented by Person Profile and Organizational Role: the latter captures the various required responsibilities and capabilities (skills) and allows those to be matched with the ones provided by the Personal Profile and required by tasks (functional operations) assigned to the person;

b) the role of humans when supporting an Enterprise Activity shall be represented by Operational Role and Person Profile: the former captures the required operational capabilities (skills) and allows those to be matched with the ones provided by the latter (Person Profile);

c) the operational role of machines shall be represented by the construct Functional Entity which captures the operational capabilities and allows those to be matched to the ones required by functional operation to which the machine has been assigned; and

d) the product role of machines shall be represented by the construct Object View which allows a selection from the Product of those attributes that are needed to describe input and output of the activities to be executed to change the state of the product.

5.7 Complementary concepts

The construct templates of Clause 6 make use of several terms for conceptual representations that are not themselves constructs, but are complementary and have a particular significance and semantics for the purposes of enterprise modelling. These complementary concepts, each of which is defined in Clause 3, are as follows:

— behavioural rule (see 6.3.5);

— constraint;

— declarative rule;

— functional operation (see 6.4.5);

— integrity rule;

— objective; and

— performance indicator.

Their occurrences in Clause 6 (except when they are used in the templates) are flagged in italics. Their usages in each of the construct templates (generally in more than one) are described in Clause E.3, where they are also flagged in italics.

6 The modelling language constructs

6.1 Overview of constructs

6.1.1 Modelling language constructs and their application areas

The modelling language constructs defined in this International Standard can be categorized according to their main application area into

— Function- and process-related: Domain, Business Process, Enterprise Activity, Event,

— Information-related: Enterprise Object, Enterprise Object View, Order, Product,

— Resource-related: Capability, Operational Role, Resource, Functional Entity, and

— Organization-related: Person Profile, Organizational Role, Organization Unit, and Decision Centre.

NOTE 1 The Organizational Unit is an organizational-entities-grouping concept. The Organizational Role is a responsibility and authority-grouping concept (i.e. assign responsibility and authority of a group of resources to one manager for example). Decision Centre is a decision-grouping (activity) concept. Decisions in Decision Centre(s) are taken by persons represented by Organizational Role(s) and the concerned Organizational Unit(s). Decisions taken in one Decision Centre can concern one or several Organizational Units. One manager can be responsible for taking decisions in one or several Decision Centres.

NOTE 2 The relations between constructs are demonstrated in Annex C, based on UML modelling concepts, and in Annex D, which shows relationships to the metamodel of modelling concepts in ISO/IEC 15414 and those developed in the European project ATHENA.

6.1.2 Purpose and applicability

This clause defines, describes and provides templates for generic modelling language constructs. These constructs form a modelling language with which a wide range of Business Process-based enterprise models can be created. Each template shall include the attributes that are pre-defined in this International Standard. Other attributes may be added to meet particular user needs (see 5.2 and 5.5) by a suitable extension of the template. Clause E.4 contains an example of order processing showing the use of the templates in enterprise model development.

Constructs are often applicable to several model views (as is evident in Annexes C and E) and are elaborated to different degrees at each of the modelling phases. Where the applicability of a construct, as defined in this International Standard, is limited to a certain model view or enterprise model phase, this is noted in remarks. Clause E.2 also shows how constructs and their templates may be used at different enterprise model phases.

NOTE Because of the interrelationships and dependencies between constructs, they cannot be specified in this International Standard in strict linear sequence (i.e. with no forward reference to not-yet-specified constructs). The sequence of text used herein has been chosen to minimize this forward reference problem and is as follows:

— start with Domain (the scope and boundary of the model);

— add functionality in terms of Business Processes, Enterprise Activities and Events;

— add information representation in terms of Enterprise Object and Object View, and the specializations Product and Order;

— provide for resourcing with Resource, Capability, and Functional Entity; and

— lastly, address organizational and role issues with Organizational Unit, Decision Centre, Person Profile, Operational Role and Organizational Role.

6.1.3 Notation used to describe template contents

Attributes are described with a combination of denotations and textual descriptions.

The notation used for this clause is that denotations are enclosed in angle brackets < and > and are either self-explanatory (e.g. <integer>) or explained in accompanying text (e.g. <item>, where item is …). Literals in all upper case denote themselves as in DM, BP, EV, NIL, PHYSICAL, etc. The literal characters :, /, (, and) are denoted by ":", "/", "(" and ")" respectively. Where more than one choice is possible for a denotation, these are separated by | (read as *or*), as in <a> | <b> | <c>. Square brackets [and] are used for convenience as a grouping container for one or more different items as in [<duration> <qual>] for the DURATION attribute of an Enterprise Activity.

Textual descriptions contain a brief definition of that attribute and, where it is felt appropriate, information providing additional details (shown in italics).

A non-empty list of <item> with an appropriate separator is written as [<item>]+, while a similar possibly empty list is written as [<item>]•. Within a list denotation, <item> may be further decomposed as in [<id> <multiplicity>]+ or [<measure> | <metric>]•.

NOTE More formally, the syntax[7] for lists is:

— list = non-empty list | possibly-empty list;

— non-empty list = item | item, separator, non-empty list; (* for convenience, abbreviated to [<item>]+ *)

— separator = ";";(* any equivalent separator character may be used instead of semicolon *)

— possibly-empty list = NIL | non-empty list; (* for convenience, abbreviated to [<item>]• *)

EXAMPLE

From the Domain template, a Design Authority with values of Organization Role and possibly also Organization Unit would be expressed as [[<identifier> "/" <name>] [NIL|":"<identifier> "/" <name>]]

From the Capability template, *data_store; id_1006453* would be valid for "[<id>]+, where <id> is one of <identifier> | <name> of Capability instances". (*NIL* would not be valid.)

From the Enterprise Activity template, *NIL* or *max_load = 10Kg; max_temp =100C* would be valid for "[<constraint>]• imposed on the Enterprise Activity instance".

Some further examples follow (two- or three-letter codes refer to construct label, followed by attribute name), progressing from simpler to more complex cases:

EO: Nature of Object	PHYSICAL | INFORMATION
CA: *Capabilities and capability attributes*	
CA: Included Capabilities	[<Capability instance>]•
CA: Function Related	[<capability attribute>]•
CA: Entity Related	[<capability attribute>]•
DM: Objectives	[<objective>]+, strategic and operational business objectives of the Domain instance
EV: Object Views	[<origin> ":" <identifier> "/" <name>]• of Object View instance defining information associated with occurrences of this Event instance
EV: Generated_by	[<origin> ":" <identifier> "/" <name>]+ of the source of this Event instance
BP: Object View Inputs	[<origin> ":" <identifier> "/" <name>]+ of Object View instances, occurrences of which can be received by occurrences of the Business Process instance]
EA: Ending Statuses	[<value> <priority>]+, ending status values produced by occurrences of this Enterprise Activity instance and their significance, where <value> is a mandatory 0-argument predicate and <priority> is an integer in a range <min, max> where min and max are integers representing the lowest and highest priorities respectively. By default, highest priority
PR: Related_to	[<identifier> "/" <name> <multiplicity>]•, defining the Enterprise Object instances that are related to this Product instance, qualified by a multiplicity which is one of [0..*] (only for early modelling phases) or [1..1] or [1..n] or [m..n]

7) See Clause A.2 for a description of the syntax notation used.

6.2 Domain

6.2.1 Purpose

A Domain represents the boundary and the content of an enterprise or a portion of an enterprise for which an enterprise model is to be created.

6.2.2 Description

Domains shall describe the parts of the enterprise to be modelled and their relationships with the external environment from a high-level management-oriented point of view. The model is subject to a given set of strategic and operational business *objectives* and *constraints*, and the purposes of the modelling. The relations between strategic and operational *objectives* should be made visible as much as possible. The Domain inputs and outputs and their origins and destinations define the domain boundaries and the relations between inputs and outputs identify the required functionalities – the business processes of the domain.

6.2.3 Usage

The construct Domain is used in all model phases. Its template shall enable

a) the capturing of all inter- and intra-organizational-related information relevant for strategic planning and decision support, and

b) its presentation in a form that is understandable by both humans and machines.

The descriptive attributes and relationships are described at a level of detail appropriate to the enterprise model phase applicable to the Domain construct's usage. The usage of an Object View or Event implies that the associated Enterprise Object is also defined at the model phase and at an appropriate level of detail.

6.2.4 Construct template for Domain

Header	
Construct label	DM
Identifier	<model-unique string>
Name	[<adjective> <noun> | <noun>], the name of Domain, where <noun> indicates the functionality or purpose, and <adjective> optionally indicates the scope
Design Authority	[[NIL | ":" <identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	short textual description of the Domain
Business Process Descriptions	[<name> | <name> <short textual description>]+, name and optionally short description of the main functionality needed to transform Domain inputs into Domain outputs *The Business Processes identified for a Domain construct are those required to achieve the Domain objectives*
Objectives[8)]	[<objective>]+, strategic and operational business objectives of the Domain instance
Constraints	[<constraint>]• imposed on the Domain instance
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
Domain Characterization	short textual description of the mission, vision and values of the Domain

8) Derived objectives occur in Business Processes, Enterprise Activities, and Decision Centres as well; consistency has to be assured.

Domain Operation	short textual description of the strategies, policies, operational concepts, business plans of the Domain
Decisional Authority	
Decision Function	[<verb> <noun> | <noun>]•, name of the decision function or functions that belong to the Domain. *A decision function is a set of decision activities or decision centres having the same category of subjects being handled, such as "manage resources", "manage products" and "plan production".*
Decision Level	[<horizon> <period>]•, where <horizon> and <period> are specified in terms of days, weeks or years. *<horizon> specifies the time interval upon which a decision taken stands and <period> specifies the reviewing frequency at the end of which a decision taken is to be revised.*
A2.2 applicable at requirement definition and later phases	
Performance Indicators	[<metric> | <measure>]•, constraint or a textual string identifying a method. *Performance indicators are defined by identifying the required value or method by which achievement of the objectives can be assessed.*
B1 Relationships relevant for all enterprise model phases	
Inputs/Outputs	
Object View Inputs	[<origin[9)] > ":" <identifier> "/" <name>]+ of Object Views, instances of which are available for occurrences of the Domain instance
Event Inputs	[<origin> ":" <identifier> "/" <name>]+ of all Events[10)], instances of which can be received by occurrences of the Domain instance
Object View Outputs	[<identifier> "/" <name> ":"<destination[11)] >]+ of all Object Views, instances of which are made available by occurrences of the Domain instance
Event Outputs	<identifier> "/" <name> ":"<destination>]+ of all Events, instances of which can be generated by occurrences of the Domain instance
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Business Processes	[[NIL | <domain> ":"] <identifier> "/" <name>]+ of Business Processes contained in this or a designated Domain instance[12)]
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

9) Here and in the following templates, <origin> denotes the Domain, Business Process or Enterprise Activity that is the source of the input. In the situation that the <origin> is an external Domain, only the name of the Domain is stated and its contents are not further defined.

10) It is the responsibility of the tool-supporting model development and maintenance to ensure that the Events listed as inputs or outputs are compatible with the Generated_by attributes of Event.

11) Here and in the following templates, <destination> denotes the Domain, Business Process or Enterprise Activity that is to receive the output.

12) It is the responsibility of the tool-supporting model development and maintenance to ensure that the Business Processes listed here are compatible with the Where_used attribute of Business Process.

6.3 Business Process

6.3.1 Purpose

The Business Process represents all or part of the domain functionalities, its internal structure and its dynamic behaviour.

6.3.2 Description

A Business Process construct shall describe the functionalities needed to produce a desired result that satisfies one or more business *objectives* derived from business *objectives* defined for the enterprise domain. This result emerges from transformations or combinations, or both, of entities into new entities or into new states which require appropriate control and functional capacity.

The internal structure of a Business Process shall be described in terms of a set of functionalities, decomposed according to modelling criteria (reduction of complexity) and the needs of operational monitoring and control. These internal functionalities shall be characterized as combinations of constituent Business Processes or Enterprise Activities, or both, and their interconnections, arranged by ordering relationships and dependencies described by *behavioural rules*, which capture the process dynamics.

The result of a Business Process shall be observable or quantifiable[13)]. It can result in material entities (such as industrial products), or information entities (such as orders, documents or data), or newly designed processes, or can be defined as the achievement of one or more designated *objectives*.

6.3.3 Usage

The construct Business Process is used in all model phases. Its template shall enable

a) the capturing of all process-related information relevant for model-based planning and decision support, and monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

The construct shall identify all constituent Business Processes or Enterprise Activities, or both, which represent the decomposition of the Business Process functionality into a number of ordered transformation actions, thus capturing relevant intermediate conditions of the entities to be changed. The construct shall describe in its *behavioural rule* set attribute the logical binding of its Business Processes or Enterprise Activities, or both, to the *behavioural rules* connecting these intermediate conditions, i.e. the logical sequence of transformation functions.

6.3.4 Construct template for Business Process

Header	
Construct label	BP
Identifier	<model-unique string>
Name	[<adjective> <noun> | <noun>], name of Business Process instance, where <noun> indicates the scope of the Business Process, and <adjective> optionally qualifies the Business Process instance
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance

13) However, in contrast with an Enterprise Activity, the template for a Business Process does not contain an ending status. Process completion is signalled by a behavioural rule FINISH action and possible exception.

Body	
A1 Descriptives relevant for all enterprise model phases	
Description	textual description of functionalities and desired result
Objectives	[<objective>]+, strategic and operational business objectives to be fulfilled by the Business Process instance
A.2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Constraints	[<constraint>]• imposed on the Business Process instance
Performance Indicators	[<metric> | <measure>]• by which achievement of the objectives can be assessed
Declarative Rules	[<rule>]•, declarative rules applicable to this Business Process instance
Behavioural Rule Set	[<behavioural rule>]+, expressed using the syntax as defined in 6.3.5 and Annex A.
A.2.3 applicable at design specification and later phases	
Priority[14)]	<integer> in a range <min, max> where min and max are integers representing the lowest and highest priorities respectively
B1 Relationships relevant for all enterprise model phases	
Where_used	[<identifier> "/" <name>]• of the Domain employing this Business Process instance
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Part_of	[<identifier> "/" <name>]• of Business Process instances in which this particular instance is involved in an aggregation. *The name of the Business Process instance being defined cannot be used in the list*
Consists_of	[<identifier> "/" <name>]+ of all Business Process and Enterprise Activity instances of which this Business Process instance is the aggregate
Inputs/Outputs	
Object View Inputs	[<origin> ":" <identifier> "/" <name>]+ of Object Views, instances of which are available for occurrences of the Business Process instance]
Event Inputs	[<origin> ":" <identifier> "/" <name>]+ of all Events[15)], instances of which can be received by occurrences of the Business Process instance
Object View Outputs	[<identifier> "/" <name> ":"<destination>]+ of all Object Views, instances of which are made available by occurrences of the Business Process instance]
Event Outputs	[<identifier> "/" <name> ":"<destination>]+ of all Events, instances of which can be generated by occurrences of the Business Process instance
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

14) Priority attributes are used at run-time to select which Business Process to activate first in the case of more than one Business Process being initiated by a Behavioural Rule set. They are also used to guide allocation of Resources to Enterprise Activities.

15) It is the responsibility of the tool-supporting model development and maintenance to ensure that the Events listed as Inputs or Outputs are compatible with the Generated_by attributes of Event.

6.3.5 Behavioural rules

6.3.5.1 Purpose

The purpose of the *behavioural rules* is

a) to identify the start of the main Business Process, and

b) describe the logical sequencing relationships of constituent Business Processes or Enterprise Activities, or both.

6.3.5.2 Description

The behaviour of a Business Process shall be described in its behavioural rule set attribute by a set of *behavioural rules* which control the sequence of constituent Business Processes and Enterprise Activities. Annex A provides a detailed description and syntax for these rules.

NOTE In contrast, the specification of the internal behaviour of an Enterprise Activity (the sequencing of its set of *functional operations*) is considered to be an implementation issue beyond the scope of this International Standard.

The logical sequencing may be more or less deterministic, corresponding to:

— well-structured processes, i.e. processes for which the expected result is known and the sequence of constituent Business Processes or Enterprise Activities, or both, is completely defined (and necessarily deterministic);

— semi-structured processes, i.e. processes for which the expected result is known and the sequence of constituent Business Processes or Enterprise Activities, or both, will be known only at run-time (semi-deterministic); and

— ill-structured processes, i.e. processes for which neither the result nor the sequence of constituent Business Processes or Enterprise Activities, or both, is completely known (non-deterministic).

6.3.5.3 Usage

The concept of *behavioural rules* is used in the requirements definition phase and later phases. The rules shall enable

a) the capturing of all the conditions that control the sequencing and the dynamic behaviour of Business Processes, and

b) their presentation in a form that is understandable by both humans and machines.

6.4 Enterprise Activity

6.4.1 Purpose

An Enterprise Activity represents the part of process functionality that is needed to realize a basic task within a Business Process of an enterprise domain. Enterprise Activities are the lowest level constituents of a Business Process (see 6.3) defined according to required user objectives for operational monitoring and control.

6.4.2 Description

An Enterprise Activity construct shall identify all things required for and produced by the execution of a particular task, which transforms function input(s) into function output(s) using control, operational role and resource inputs and optionally producing control, operational role and resource-related information outputs.

The latter shall contain status information on the execution of the task as they relate to the activity itself and the resource, respectively. Operational Role inputs identify required skills to be provided by assigned Person Profiles. Resource inputs shall identify the required and provided capabilities, respectively.

6.4.3 Usage

The construct Enterprise Activity is used in the requirements definition phase and later phases. Its template shall enable the capturing of all relevant information (inputs and outputs) needed and created in

a) the execution of the identified functionality, and

b) model-based planning and decision support, and monitoring and control of operational processes.

Such information should be presented in a form that is understandable by both humans and machines. In addition, the template shall identify all constituent *functional operations*, which represent the decomposition of the Enterprise Activity functionality and are needed for Resource and Operational Role assignments.

The Enterprise Activity required capability should be matched by the Capability of the Resources that should be supplied.

6.4.4 Construct template for Enterprise Activity

Header	
Construct label	EA
Identifier	<model-unique string>
Name	[<verb> <adjective> <noun> | <verb> <noun> | <verb>], name of Enterprise Activity instance, where <verb> characterizes the imperative nature of the Enterprise Activity, <adjective> and <noun> are optional qualifier and scope indications respectively of the Enterprise Activity
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance.
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Description	textual description
Activity Behaviour	textual description of the algorithm[16)] specifying the control flow of the functional operations of which this Enterprise Activity is comprised – see 6.4.5
Objectives	[<objective>]+, operational business objectives to be fulfilled by the Enterprise Activity instance
Constraints	[<constraint>]• imposed on the Enterprise Activity instance
Performance Indicators	[<metric> | <measure>]• by which achievement of the objectives can be assessed

16) The control flow of these subtasks, collectively called activity behaviour, can be defined in an algorithm which describes the input/output conditions for sequence control of the Functional Entities involved in the execution of the Enterprise Activity. The specification of the algorithm is considered to be an implementation issue beyond the scope of this International Standard.

A2.3 applicable at design specification and later phases	
Consists_of	[<functional operation> | <functional operation> "(" [<param>]+ ")"]+, instances of which this Enterprise Activity instance is the aggregate[17)] *Each functional operation is identified by a functional operation name and its input and output arguments; this name is to be unique within the scope in which it is defined and used*
Inputs/Outputs	
Ending Statuses	[<value> <priority>]+, ending status values produced by occurrences of this Enterprise Activity instance, where <value> is a mandatory 0-argument predicate and <priority>[18)] is an integer in a range <min, max> where min and max are integers representing the lowest and highest priorities respectively. By default, highest priority
Duration	[<duration> <qual>]•, attribute pairs defining the duration of occurrences of the Enterprise Activity instance, where <qual> is a code representing one of 'average', 'minimum', 'maximum', 'actual'
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Where_used	[<identifier> "/" <name>]+ of Business Process employing this Enterprise Activity
Inputs/Outputs	
Function Inputs	[<origin> ":" <identifier> "/" <name>]+ of Object View instances describing input information to be processed by this Enterprise Activity instance
Control Inputs	[<origin> ":" <identifier> "/" <name>]+ of Object View instances describing input information providing run-time data used but not modified by occurrences of this Enterprise Activity instance
Required Operational Roles	[<identifier> "/" <name>]+ of Operational Roles defining required operational capabilities of occurrences of this Enterprise Activity instance *Operational Role inputs identify required skills to be provided by Person Profiles (see 6.15).*
Required Capabilities	[<identifier> "/" <name>]+ of Capability defining required capabilities of occurrences of this Enterprise Activity instance
Function Outputs	[<identifier> "/" <name> ":"<destination>]+ of Object View instances describing the information outputs produced by occurrences of this Enterprise Activity instance *The function outputs contain status information on the execution of the task and subtasks as they relate to the activity and associated resources.*
Input Events	[<origin> ":" <identifier> "/" <name>]+ of all Events[19)], instances of which can be received by occurrences of the Enterprise Activity instance
Output Events	<identifier> "/" <name> ":"<destination>]• of all Events, instances of which can be generated by occurrences of the Enterprise Activity instance
B2.3 applicable at design specification and later phases	
Inputs/Outputs	
Operational Role Outputs	[<identifier> "/" <name> ":"<destination>]• of Object View instances describing the relevant information for Operational Roles after the execution of occurrences of this Enterprise Activity instance

17) The way in which the functional operations required by an Enterprise Activity are matched to the Functional Entities that can execute them is an implementation issue outside the scope of this International Standard.

18) As for Business Processes, priority is used at run-time to select which Business Process or Enterprise Activity to initiate first when the completion times of two Enterprise Activities are complete within the same interval of observable time.

19) It is the responsibility of the tool-supporting model development and maintenance to ensure that the Events listed as inputs or outputs are compatible with the Generated_by attributes of Event.

Control Outputs	[<identifier> "/" <name> ":"<destination>]• of Object View instances describing the status information of the Enterprise Activities after the execution of occurrences of the Enterprise Activity instance
Resource Inputs	[<origin> ":" <identifier> "/" <name>]+ of Resource instances required by occurrences of this Enterprise Activity instance[20)] *Resource inputs identify the provided capabilities (see 6.10 and 6.11)*
Resource Outputs	[<identifier> "/" <name> ":"<destination>]• of Object View instances describing the status information of Resources after the execution of occurrences of the Enterprise Activity instance
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.4.5 Functional operation

6.4.5.1 Purpose

A *functional operation* represents a part of the functionality of an Enterprise Activity, which has been decomposed into a number of ordered transformation functions.

6.4.5.2 Description

To enable a consistent person and resource assignment, the functionality of the Enterprise Activity represented by the required Operational Roles and Capabilities shall be partitioned into *functional operations* in such a way that they are executable by an autonomous resource that is described by the Functional Entity construct. Each *functional operation* shall be defined in the operation set of one or more Functional Entities.

6.4.5.3 Usage

The concept of *functional operation* is used in the design specification phase and later phases.

NOTE 1 The control flow of these subtasks, collectively called activity behaviour, can be defined in an algorithm[21)], which describes the input/output conditions for sequence control of the Functional Entities involved in the execution of the Enterprise Activity. The specification of the algorithm is considered to be an implementation issue beyond the scope of this International Standard.

NOTE 2 This International Standard does not specify a language or form of representation for the control flow algorithm.

6.5 Event

6.5.1 Purpose

The purpose of the Event construct is to capture reason, origin and destination of an event.

20) It is the responsibility of the designer with supporting tools to ensure that these Resources include at least those specified as able to provide the required capabilities.

21) This is in contrast to Business Processes whose behaviour is specified by Behavioural Rules as specified in 6.3.3.

6.5.2 Description

An Event shall represent the initiation of a state change in the enterprise or its environment, to be used to initiate the execution of one or more processes, and hence shall activate Business Processes or Enterprise Activities, or both, by initiating the processing of the behavioural rule set associated with a Business Process (see also 6.3.5).

An Event is also used to signal an occurrence of significance to a Domain.

6.5.3 Usage

The construct Event is used in all model phases. Its template will capture the event-relevant information and present it in a form that is understandable by both humans and machines.

NOTE Some Events might simply be signals that something has occurred while others have associated information that will be conveyed by an Object View, e.g. representing an Order (see the example in E.4.2.5).

EXAMPLE Event: arrival of customer order (customer order information).

Every Event is a unique occurrence. However, Events shall be given a name, which may be qualified by an adjective, to indicate some specific structure and behaviour.

6.5.4 Construct template for Event

Header	
Construct label	EV
Identifier	<model-unique string>
Name	[<adjective> <noun> | <noun>], name of the Event instance, where <noun> indicates the entity causing the Event, and <adjective> is an optional qualifying attribute
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance.
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	textual description
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
	Not applicable
A2.3 applicable at design specification and later phases	
Timestamp	<yy.mm.dd:hh.mm.ss:mmm>, data type defining date and time to be provided at occurrence creation time. *Timestamp has a fixed value, which is a data type for date and time (and cannot be changed because it is known by the system).*
Priority[22)]	<integer> in a range <min>, <max> where min and max are integers representing the lowest and highest priorities respectively. By default, highest priority
B1 Relationships relevant for all enterprise model phases	
Object Views	[NIL|<origin> ":" <identifier> "/" <name>]• of Object View instance defining information associated with occurrences of this Event instance
Generated_by	[<origin> ":" <identifier> "/" <name>]+ of the source of this Event instance

22) As with Business Processes, Priority is used at run-time to select which Business Process or Enterprise Activity to initiate first when near-simultaneous Events occur.

Initiates	[<identifier> "/" <name> ":"<destination>]+ of the destination that can receive this Event instance[23)]
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.6 Enterprise Object

6.6.1 Purpose

The purpose of the Enterprise Object construct is to describe those characteristics (attributes and relational structure) and to provide for selection of relevant parts (Object Views) that are to be identified in the modelling process and used during the operational phase.

6.6.2 Description

An Enterprise Object shall represent the common characteristics from an information viewpoint of a thing (an enterprise entity) as it exists during its lifetime. Its usage shall be restricted to those situations where only the information aspects of the entity under consideration are relevant.

6.6.3 Usage

The construct Enterprise Object is used in all model phases. Its template shall enable

a) the capturing of all relevant information and relations that are needed for model-based planning and decision support, monitoring and control of operational processes and their execution, and

b) its presentation in a form that is understandable by both humans and machines.

The entry "Nature of Object" shall be used to express an Enterprise Object's primary characteristic that determines how easily it can be transported, replicated, etc. The term is to be used to differentiate the information flow from the physical flow among Enterprise Activities.

6.6.4 Construct template for Enterprise Object

Header	
Construct label	EO
Identifier	<model-unique string>
Name	name of the Enterprise Object instance

23) It is the responsibility of the tool-supporting model development and maintenance to ensure that the Generated_by attributes are compatible with the Origin/Destination Event lists of Domain, and the Input/Output lists of Business Process and Enterprise Activity.

Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name> = <property_value>]+, elements representing properties and their values for the entity represented by the Enterprise Object instance
Constraints	[<constraint>]• imposed on selected named attributes of the Enterprise Object instance[24)]
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Integrity Rules	[<integrity rule>]• applicable to attributes of the Enterprise Object instance in the requirements definition phase
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Enterprise Object relationships in the form:	
Specialization_of	[<identifier> "/" <name>]• of Enterprise Objects that are generalizations of this particular instance[25)]. *The name of the Enterprise Object instance being defined cannot be used in the list*
Part_of:	[<identifier> "/" <name>]• of Enterprise Object instances in which this particular instance is involved in an aggregation. *The name of the Enterprise Object instance being defined cannot be used in the list*
Consists_of:	[<identifier> "/" <name>]• of all Enterprise Object instances of which this Enterprise Object instance is the aggregate. *The name of the Enterprise Object instance being defined cannot be used in the list*[26)].
Related_to	[<identifier> "/" <name > <multiplicity>]•, defining the Enterprise Object instances that are related to this Enterprise Object instance, and where multiplicity is one of [0..*] (only for early modelling phases) or [1..1] or 1..*] or [m..n].
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

24) It is the responsibility of the tool-supporting model development and maintenance to ensure that attributes referred to in constraints and integrity rules are consistent with the attributes as defined for the construct instances involved.

25) This implies that this Enterprise Object inherits all descriptives and relationships from its generalization(s).

26) Here, as for other Constructs where both Part_of and Consists_of are used, it is accepted that these are dual relationships; depending on the purposes of the modeller, either or both may be used. In the latter case, it is the responsibility of the modelling tool to ensure the consistency of selected attributes.

6.7 Enterprise Object View (Object View)

6.7.1 Purpose

The purpose of an Object View construct is to enable the identification of relevant attributes from a particular Enterprise Object as required by a Domain, a Business Process, or an Enterprise Activity for the definition of their inputs and outputs.

6.7.2 Description

An Enterprise Object View shall represent a subset of the descriptive attributes (including associated *constraints* and *integrity rules*[27)]) of an Enterprise Object. Object Views shall also be used to convey any information that may be associated with an Event including lists identifying Enterprise Objects for which Organizational Units have responsibility or authority, or both, and which are affected by decisions made by Decision Centres.

6.7.3 Usage

The construct Object View is used in all model phases. Its template shall enable

a) the capturing of relevant information needed for strategic planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

NOTE The term “Enterprise Object View” is usually shortened to “Object View”.

6.7.4 Construct template for Object View

Header	
Construct label	OV
Identifier	<model-unique string>
Name	name of the Object View instance
Design Authority	[[<identifier> “/” <name>] [NIL | “:” <identifier> “/” <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	short textual description
Properties	[<property_name><property_value>]+, elements representing properties and their values of the Object View instance (a selection of the attributes of the Enterprise Object instance to which the Object View instance applies)
Constraints	[<constraint>]• imposed on selected attributes of the Enterprise Object instance from which the Object View instance is derived
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Integrity Rules	[integrity rule]• applicable to the selected attributes of the Enterprise Object instance from which the Object View instance is derived
B1 Relationships relevant for all enterprise model phases	
Enterprise Object	[<identifier> “/” <name>] of the Enterprise Object instance from which the Object View instance is derived

27) These are contained in Part A of the body of the template. It is the responsibility of the tool-supporting model development and maintenance to ensure that constraints and integrity rules are consistent.

Events	[<identifier> "/" <name>]• of Event instance associated with this Object View instance
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.8 Product

6.8.1 Purpose

The purpose of the Product construct is to describe all intermediate stages of product life cycles with regard to both material and informational aspects.

6.8.2 Description

A Product shall represent the products and specializations thereof, whose fabrication and sale is the aim of the enterprise. A Product is a specialization of the Enterprise Object construct for which *integrity rules* are valid for all phases, and having a distinguishing construct label to reflect its purpose and usage.

By particularization and specialization of attributes, Product subclasses shall be defined to meet the specific requirements of a certain branch of industry or particular enterprise. Complex product structures shall be represented by Part_of relationships between well-distinguished Product subclasses.

6.8.3 Usage

The construct Product is used in all model phases. Its template shall enable

a) the capturing of all product-related information relevant for strategic planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

Depending on the level of detail, all Product stages relevant to the manufacturing process shall be described as inputs/outputs of Enterprise Activities and their logical sequence representing the processing of the Product.

6.8.4 Construct template for Product

Header	
Construct label	PR
Identifier	<model-unique string>
Name	name of the Product instance
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance

Body	
A1 Descriptives relevant for all enterprise model phases	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name><property_value>]+, elements representing properties and their values of the Product instance as for Enterprise Object
Constraints	[<constraint>]• imposed on attributes of the Product instance.
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Integrity Rules	[<integrity rule>]• applicable to attributes of the Product instance[28)].
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Product relationships in the form:	
Specialization_of	[<identifier> "/" <name>]• of Product classes which are a generalization of this particular instance. *The name of the Product instance being defined cannot be used in the list*
Part_of:	[<identifier> "/" <name>]• of Product instances in which this particular instance is involved in an aggregation. *The name of the Product instance being defined cannot be used in the list*
Consists_of:	[<identifier> "/" <name>]• of all Product instances of which this Product instance is the aggregate. *The name of the Product instance being defined cannot be used in the list*
Related_to	[<identifier> "/" <name> <multiplicity>]•, defining the Enterprise Object instances that are related to this Product instance, qualified by a multiplicity which is one of [0..*] (only for early modelling phases) or [1..1] or 1..n] or [m..n]
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.9 Order

6.9.1 Purpose

The purpose of an Order construct is to describe what has to be done, which products are to be produced, which resources are to be used, and what purchases have to be made.

28) Here, as for other Constructs where both constraints and integrity attributes are used, it is accepted that these are dual relationships; depending on the purposes of the modeller, either or both may be used. In the latter case, it is the responsibility of the modelling tool to maintain consistency.

6.9.2 Description

An Order shall represent an instruction from one authority to other authorities for performance of an operation. An Order is a specialization of the Enterprise Object construct for which *integrity rules* are valid for all phases, and which has a distinguishing construct label to reflect its purpose and usage.

6.9.3 Usage

The construct Order is used in all model phases. Its template shall enable

a) the capturing of all order-related information relevant for strategic planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

The Order shall be as exact as necessary, i.e. all information for the execution of the operation that cannot be supplied by the responsible Organizational Unit shall be provided.

There are different possibilities to describe what is expected and how this is to be achieved, such as by sending an Order to the Organizational Unit that is responsible for the correct execution of the Business Process. Within an enterprise, many different kinds of Orders with different associated information may be defined (e.g. customer order, manufacturing order, etc.).

NOTE An association of an Order with an Event is represented by an Object View, as described in E.4.2.1 and E.4.2.5.

6.9.4 Construct template for Order

Header	
Construct label	OR
Identifier	<model-unique string>
Name	name of the Order instance
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name><property_value>]+, elements representing properties and their values of the Order instance as for Enterprise Object
Constraints	[<constraint>]• imposed on attributes of the Order instance
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Integrity Rules	[<integrity rule>]• applicable to attributes of the Order instance[29].
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Order relationships in the form:	
Specialization_of	[<identifier> "/" <name>]• of Order classes that are a generalization of this particular instance. *The name of the Order instance being defined cannot be used in the list*

29) Here, as for other Constructs where both constraints and integrity attributes are used, it is accepted that these are dual relationships; depending on the purposes of the modeller, either or both may be used. In the latter case, it is the responsibility of the modelling tool to maintain consistency.

Part_of:	[<identifier> "/" <name>]• of Order instances in which this particular instance is involved in an aggregation. *The name of the Order instance being defined cannot be used in the list*
Consists_of:	[<identifier> "/" <name>]• of all Order instances of which this Order instance is the aggregate. *The name of the Order instance being defined cannot be used in the list*
Related_to	[<identifier> "/" <name > <multiplicity>]•, defining the Enterprise Object instances that are related to this Order instance (such as Product), qualified by a multiplicity which is one of [0..*] (only for early modelling phases) or [1..1] or [1..n] or [m..n]
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.10 Resource

6.10.1 Purpose

The purpose of the Resource construct is to classify and describe in terms of capabilities all material and informational aids in an enterprise, such as machining equipment, tools and facilities, equipment for data processing, documents and files containing geometrical, material and informational characteristics, which are required to execute, enable or support the execution of Enterprise Activities in an enterprise.

6.10.2 Description

A Resource shall represent some or all of the capabilities provided for an Enterprise Activity according to its required capabilities, where these capabilities are those of any device, tool or means[30)] at the disposal of the enterprise to produce goods or services. A Resource is a specialization of Enterprise Object that specializes the attribute properties and adds new attributes:

— provided operational roles, and

— provided capability.

The resulting construct template has a distinguishing construct label to reflect its purpose and usage.

For each kind of Resource, all relevant qualities and services of that Resource shall be described in terms of a Capability (see 6.11), related to its ability to complete *functional operations* in the enterprise and its availability for, and *constraints* on, carrying out those tasks. The corresponding functions for acquiring or preserving the ability and the availability respectively (i.e. preparation, provision, servicing), as well as the logical sequence of these functions, should also be described.

30) In this context "means" includes software and datasets, such as a STEP file. It does not include human resources, which are instead represented by Personal Profiles and specializations thereof.

6.10.3 Usage

The construct Resource is used in the design specification phase and later phases. Its template shall enable

a) the capturing of all resource-related information relevant for planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

The description of 6.10.2 also implies some possible subclasses of Resource that have to be specialized and specified for a particular enterprise.

6.10.4 Construct template for Resource

Header	
Construct label	RE
Identifier	<model-unique string>
Name	<name> of the Resource instance
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority and responsibility to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
	Not applicable
A2.3 applicable at design specification and later phases	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name><property_value>]+, elements representing properties and their values of the Resource instance *These properties define necessary descriptive characteristics other than those that are task-related*[31]*; example property_names might be location, when installed, book value*
Constraints	[<constraint>]• imposed on the above attributes of the Resource instance
Integrity Rules	[<integrity rule>]• applicable to the above attributes of the Resource instance
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Provided Operational Roles	[NIL | <identifier> "/" <name>]• of the Operational Roles providing human capabilities to operate this Resource instance
Provided Capability	[<identifier> "/" <name>] of the Capability defining the provided capabilities of this Resource instance. *Resource (unlike a Functional Entity, see 6.12) provides only a single capability*

31) Attributes that are task-related are contained within the associated Capability.

Class Relationships – list of Resource relationships in the form:	
Specialization_of:	[<identifier> "/" <name>]• of Resource classes which are a generalization of this particular instance. *The name of the Resource instance being defined cannot be used in the list*
Part_of:	[<identifier> "/" <name>]• of Functional Entity instances of which this Resource is involved in an aggregation] *The name of the Resource instance being defined cannot be used in the list*
Consists_of:	[<identifier> "/" <name>]• of Resource instances and Functional Entity instances of which this Resource instance is the aggregate. *The name of the Resource instance being defined cannot be used in the list*
Related_to	[<identifier> "/" <name > <multiplicity>]•, defining the Enterprise Object instances that are related to this Resource instance, qualified by a multiplicity which is one of [0..*] (only for early modelling phases) or [1..1] or [1..n] or [m..n]
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to change the operation of this instance

6.11 Capability

6.11.1 Purpose

A Capability represents the elements of both the capabilities required by an Enterprise Activity (see 6.4) and those provided by a Resource (see 6.10).

6.11.2 Description

A Capability shall describe, by means of capability attributes and other included Capabilities, the functionality which is needed and provided to support the execution of the task to be performed by an Enterprise Activity, and to identify *constraints* determined by the things to be processed (such as safety and security aspects, certain tooling and working space dimensions, data processing/storing and main memory/storage capacity, etc.) and possibly time restrictions.

Capability attributes shall be defined in terms of a Resource-dependant attribute and a value, set of values, or permissible range of values for that attribute.

EXAMPLE Maximum/minimum part sizes or weights that can be handled, degree of precision or repeatability, etc.

6.11.3 Usage

The construct Capability is used in the requirements definition phase and later phases. Its template shall enable

a) the capturing of all Enterprise Activity- and Resource-related information relevant for model-based planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

The Capability required by an Enterprise Activity shall be matched by the provided Capability of the Resources.

6.11.4 Construct template for Capability

Header	
Construct label	CA
Identifier	<model-unique string>
Name	name of this Capability
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Description	short textual description
Function-related Attributes	[<capability attribute>]•, identifying and constraining Enterprise Activity-related attributes required by one or more Resource instances *These resource instances are listed in the Where_required attribute below.*
Constraints	[<constraint>]• -imposed on task-related attributes of one or more Resource instances
Integrity Rules	[<integrity rule>]• applicable to task-related attributes of one or more Resource instances
A2.3 applicable at design specification and later phases	
Performance-related Attributes	[<capability attribute>]• identifying and constraining performance-related attributes of one or more Resource instances
Operation-related Attributes	[<capability attribute>]• identifying and constraining Enterprise Activity-related attributes of one or more Resource instances
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Capabilities Included	[<Capability instance>]•, names of included Capability instances
Where_required	[<identifier> "/" <name>]+ of Enterprise Activities that require an instance of this Capability
Where_provided	[<identifier> "/" <name>]+ of Resources that provide instances of this Capability
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively responsible for this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively responsible for this instance

6.12 Functional Entity

6.12.1 Purpose

The purpose of Functional Entity is to describe the Operational Roles and Capabilities provided by the Functional Entity as well as the set of *functional operations* that can be assigned to it and which it can execute in a quasi-autonomous mode.

6.12.2 Description

A Functional Entity shall represent a special type of resource which is able to execute one or more *functional operations* and to be a self-sufficient part of the manufacturing system (including all needed ICT capabilities). A Functional Entity is a specialization of the Resource construct having an additional attribute (operation set) to list those *functional operations*, which are able to provide multiple capabilities, and having a distinguishing construct label to reflect its purpose and usage.

NOTE The ways in which the mapping of Functional Entity to *functional operations* is to occur, the measures to be used in making the determination of correspondence, the assurance of capacity and similar implementation issues are outside the scope of this International Standard.

6.12.3 Usage

The construct Functional Entity is used in the design specification phase and later phases. Its template shall enable

a) the capturing of all information related to humans and ressources that is relevant for model-based planning and decision support, monitoring and control of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

6.12.4 Construct template for Functional Entity

Header	
Construct label	FE
Identifier	<model-unique string>
Name	<name> of the Functional Entity instance
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
	Not applicable
A2.3 applicable at design specification and later phases	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name><property_value>]+, elements representing properties and their values for the Functional Entity instance *These define necessary descriptive characteristics (such as capacity, location, cost drivers, status)*
Operation Set	[<functional operations>]+ executable by this Functional Entity instance. *Each functional operation is identified by a functional operation name and its input and output arguments; this name is to be unique within the scope in which it is defined and used.*
Constraints	[<constraint>]• imposed on selected named attributes of the Functional Entity instance
Integrity Rules	[<integrity rule>]• applicable to attributes of the Functional Entity instance
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable

B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Provided Operational Role	[NIL | <identifier> "/" <name>]• of the Operational Roles providing human capabilities to operate this Functional Entity instance
Provided Capabilities	[<identifier> "/" <name>]• of Capability instances whose aggregate defines the provided capabilities of this Functional Entity instance. *This shall include its capability to process information for sending, receiving, processing and (optionally) storing information. A Functional Entity (unlike a Resource) can provide multiple capabilities.*
Class Relationships – list of Functional Entity relationships in the form:	
Specialization_of	[<identifier> "/" <name>]• of Functional Entity classes which are a generalization of this particular instance. *The name of the Functional Entity instance being defined cannot be used in the list*
Part_of:	[<identifier> "/" <name>]• of Resource and Functional Entity instances of which this Functional Entity is a part. *The name of the Functional Entity instance being defined cannot be used in the list*
Consists_of:	[<identifier> "/" <name>]• of Functional Entity instances of which this Functional Entity instance is the aggregate. *The name of the Functional Entity instance being defined cannot be used in the list*
Related_to	[<identifier> "/" <name > <multiplicity>]•, defining the Enterprise Object instances that are related to this Functional Entity instance, qualified by a multiplicity which is one of [0..*] (only for early modelling phases) or [1..1] or [1..n] or [m..n]
Operational Relationships	
Operation Responsibility	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having responsibility for operation of this instance
Operation Authority	[[<identifier> "/" <name>] [":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority for operation of this instance

6.13 Organizational Unit

6.13.1 Purpose

The purpose of the Organizational Unit construct is to describe an identifiable entity together with its position relative to other such entities in the enterprise organizational structure.

6.13.2 Description

Organizational Units shall represent the formal, hierarchical or administrative structure of an enterprise, or some combination thereof. Its content shall be determined by naming the assigned entities for which it has authority, responsibility, or both. Position in the hierarchy shall be represented by the definition of the assignments of lower level units to the current unit and its own assignment to higher-level units. The construct enables multidimensional organization structures (i.e. matrix organizations, network organizations and others) by allowing multiple relationships to higher-level units.

Each Organizational Unit shall contain at least one relationship to an Organizational Role specifying the required or provided organizational skills and responsibilities.

6.13.3 Usage

The construct Organizational Unit is used in all model phases. Its template shall enable the capturing of all Organization hierarchy-related information relevant for model-based planning and decision support, monitoring and control of operational processes as well as their execution and its presentation in both human- and machine-understandable form. The template shall identify all Organizational Units to which this Organizational Unit is assigned and all Organizational Units and Organizational Roles which are part of this Organizational Unit.

NOTE 1 The organizational areas that are depicted by an Organizational Unit can be of very different sizes (e.g. the whole enterprise, a department, a section or a team).

NOTE 2 Organizational structures play an important role within enterprises. The structures can describe reporting lines, profit centres, responsibilities of individuals or teams and the relationships between them.

6.13.4 Construct template for Organizational Unit

Header	
Construct label	OU
Identifier	<model-unique string>
Name	[<adjective> <noun>], name of Organizational Unit, where <adjective> qualifies the Organizational Unit, and <noun> relates to the scope of the Organizational Unit
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	textual description of the purpose of the Organizational Unit
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
	Not applicable
A2.3 applicable at design specification and later phases	
Organization Level	A textual description of the level of this Organizational Unit in relation to the organization hierarchy[32)]
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Authorities and Responsibilities[33)]	
Process-related authority/responsibilities:	[<identifier> "/" <name>]• identifying the Domains, Business Processes, Enterprise Activities, and Events for which this Organizational Unit has authority or responsibility or both

32) The consistency of Organization Level use within Organization Unit and Decision Centre is the responsibility of the implementation tool.

33) The consistency of these responsibilities and the corresponding attributes of the named constructs is the responsibility of the implementation tool.

Information-related authority/responsibilities:	[<identifier> "/" <name>]• identifying the Enterprise Objects, Object Views, Products, and Orders for which this Organizational Unit has authority or responsibility or both
Resource-related authority/responsibilities:	[<identifier> "/" <name>]• identifying the Person Profiles, Resources, Capabilities and Functional Entities for which this Organizational Unit has authority or responsibility or both
Assignments	
Assigned Organization Roles /Units	[<identifier> "/" <name>]+ identifying the Organizational Roles or Organizational Units or both that are assigned to this Organizational Unit
Assigned to Organization Units	[<identifier> "/" <name>]• identifying the Organizational Units which have authority or responsibility or both on this Organizational Unit

6.14 Decision Centre

6.14.1 Purpose

The purpose of the Decision Centre construct is to describe its contents (in terms of classes of decisional functions, e.g. manage products, manage resources, plan production, others) and its relations to other Decision Centres (the related decisional and information flow) and to Organizational Units.

6.14.2 Description

Decision Centres shall represent the decisional structure of an enterprise. A decisional structure is defined when the set of decision centres are known and their relations determined.

The Decision Centre construct shall describe identifiable entities together with their position relative to other such entities in the enterprise decision structure. The Decision Centre content shall be determined by a set of decisions (and decision-making activities) belonging to the Centre and by decision frame attributes (*objectives*, variables, *constraints*, etc.).

NOTE The details of the Decision Centre content are found in ISO 15704:2000/Amd.1:2005, Annex C.

The position of a Decision Centre in the decisional structure shall be represented by a functional decision category (manage products, manage resources, plan production, others), the decision level characterized by Horizon and Period and its relations to other Decision Centres (the related decisional and information flow).

Each Decision Centre shall contain at least one relationship to another Decision Centre and one to an Organizational Unit. The relationships between Decision Centres and with Organizational Roles and Units shall be described by means of associations in the design specification and later phases.

6.14.3 Usage

The construct Decision Centre is used in the concept definition phase and later phases. Its template shall enable

a) the capturing of all decision-related information relevant for model-based planning and decision support, monitoring and control of operational processes, as well as their execution, and

b) its presentation in a form that is understandable by both humans and machines.

6.14.4 Construct template for Decision Centre

Header	
Construct label	DC
Identifier	<model-unique string>
Name	[<verb> <adjective> <noun> | <verb> <noun> | <noun>], name of Decision Centre, where <verb> characterizes the imperative nature of the Decision Centre, <adjective> and <noun> are optional qualifier and scope indicators of the Decision Centre
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
Description	textual description of the purpose of the Decision Centre
Decision Frame (defining the decision context to be satisfied by the Decision Centre instance)	
Objectives	[<item>]+, where the items specify the objectives to be achieved
Variables	[<item>]+, where the items specify the variables involved
Constraints	[<item>]+, where the items specify the constraints to be satisfied
Decision Category	[<verb> <noun>], name of the decision function to which the Decision Centre belongs, describing the decision function. *A decision function is a set of decision activities or Decision Centres having the same category of subjects being handled, such as "manage resources", "manage products", "plan production"*
Decision Level	[<horizon> <period>] where <horizon> and <period> are specified in terms of days, weeks or years. *<horizon> specifies the time interval upon which a decision taken stands and <period> specifies the reviewing frequency at the end of which a decision taken is to be revised.*
A2.2 applicable at requirements definition and later phases	
	As A2.1 above
A2.3 applicable at design specification and later phases	
Organization Level	a textual description of the level of this Decision Centre in relation to the organization hierarchy
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Operational Applicability	
Process applicability	[<identifier> "/" <name>]•, identifying to which Domains, Business Processes, Enterprise Activities and Events decisions made by this Decision Centre apply
Information applicability	[<identifier> "/" <name>]•, identifying to which Enterprise Objects, Object Views, Products and Orders decisions made by this Decision Centre apply
Resource applicability	[<identifier> "/" <name>]•, identifying to which Person Profiles, Resources, Capabilities and/or Functional Entities decisions made by this Decision Centre apply
Assignments	
Assigned Decision Centres	[<identifier> "/" <name>]•, identifying the Decision Centres controlled by this Decision Centre

Assigned to Decision Centre	[<identifier> "/" <name>]•, specifying the Decision Centre that has responsibility and authority on this Decision Centre
Assigned to Organization Role	[<identifier> "/" <name>] of Organizational Role responsible for the Decision Centre.
Assigned to Organization Unit	[<identifier> "/" <name>]+, identifying the Organizational Unit to which this Decision Centre is assigned

6.15 Person Profile

6.15.1 Purpose

The purpose of the Person Profile construct is to describe the human skill profiles available to serve the assigned organizational and operational tasks and to fulfil the responsibilities associated with those tasks.

6.15.2 Description

Person Profiles shall represent the human skills that are actually or potentially relevant for the work of Organizational Units and Enterprise Activities. A skill profile shall be a list of predefined or user-defined human organizational and operational skills, described in terms appropriate to the user of those skills and understandable to the person providing them.

Each Person Profile shall contain at least one relationship to an Organizational Unit. The relationships between Person Profiles and Organizational Units shall be described by means of an assignment attribute in the design specification and later phases.

6.15.3 Usage

The construct Person Profiles is used in design specification and later phases. Its template shall enable

a) the capturing of all skill-related information relevant for model-based planning and decision support, monitoring, control and execution of organizational and operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

6.15.4 Construct template for Person Profile

Header	
Construct label	PPR
Identifier	<model-unique string>
Name	[<adjective> <noun> | <noun>], name of Person Profile, where <adjective> optionally qualifies the Person Profile and <noun> relates to the scope of the Person Profile
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
	Not applicable

A2.3 applicable at design specification and later phases	
Employee Profile relevant for the employment in the enterprise	
Organization-related Skills[34)]	[<skill>]•, instances of which are provided for organizational, decision-making and problem-solving tasks
Operation-related Skills	[<skill>]•, instances of which are provided for operational tasks
Role Job Description	textual description of organizational, decision-making, problem-solving or operational tasks
Provider of Skills	[<identifier> "/" <name>]•, where the <identifier> and <name> specifies the person(s) providing the skill
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
	Not applicable
B2.3 applicable at design specification and later phases	
Assignments	
Assigned to Organization Unit	[<identifier> "/" <name>]• of the Organizational Unit that has responsibility and authority on this Person Profile
Assigned to Organizational Role	[<identifier> "/" <name>]• of Organizational Roles to which this Person Profile is assigned
Assigned to Operational Role	[<identifier> "/" <name>]• of Operational Roles for which this Person Profile provides skills

6.16 Organizational Role

6.16.1 Purpose

The purpose of an Organizational Role construct is to describe the skill profile required to serve the defined organizational responsibilities and to fulfil those responsibilities. A skill profile shall be a list of predefined or user-defined human organizational skills, described in terms appropriate to the user of those skills and understandable to the person providing them.

6.16.2 Description

Organizational Roles shall represent, within a given hierarchical structure of an enterprise, the organizationally relevant human skills and responsibilities required and provided to perform those organizational responsibilities that are assigned to the particular Organizational Role.

The enterprise's hierarchical structure shall be represented by Organizational Units and Decision Centres.

Each Organizational Role shall contain at least one relationship to an Organizational Unit. The relationships between Organizational Roles and Organizational Units shall be described by means of an assignment attribute in the design specification and later phases.

6.16.3 Usage

The construct Organizational Role is used in requirements definition and later phases. Its template shall enable

a) the capturing of all Organizational responsibility-related information relevant for model-based planning and decision support, monitoring and control of operational processes, and

34) At least one of the organization- or operation-related skills shall be present.

b) its presentation in a form that is understandable by both humans and machines.

The template shall identify all Organizational Units for which this Organizational Role is required.

6.16.4 Construct template for Organizational Role

Header	
Construct label	ORR
Identifier	<model-unique string>
Name	[<adjective> <noun>], name of Organizational Role, where <adjective> qualifies the Organizational Role and <noun> relates to the scope of the Organizational Role
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Organizational Skills	[<skill>]+, instances of which are required for organizational, decision-making and problem-solving responsibilities
Responsibilities	[<responsibility>]• given to this Organizational Role *This is a textual description of organizational, decision-making or problem-solving responsibilities*
Authorities	[<authority>]• given to this Organizational Role
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Specialization_of	[<identifier> "/" <name>]• of Organizational Roles that are a generalization of this particular instance. *The name of the Organizational Role instance being defined cannot be used in the list*
B2.3 applicable at design specification and later phases	
Assignments	
Responsible Organization Unit	[<identifier> "/" <name>] of the Organizational Unit that has responsibility and authority on this Organizational Role
Assigned Decision Centre	[<identifier> "/" <name>]• of the Decision Centre which is assigned to this Organizational Role

6.17 Operational Role

6.17.1 Purpose

The purpose of an Operational Role construct is to describe the skill profile required and provided to undertake the defined operational tasks. A skill profile shall be a list of predefined or user-defined human operational skills, described in terms appropriate to the user of those skills and understandable to the person providing them.

6.17.2 Description

Operational Roles shall represent the relevant human skills and responsibilities required and provided to perform the operational tasks that are assigned to the particular Operational Role.

Each Operational Role shall contain at least one relationship to an Enterprise Activity. The relationships between Operational Roles and Enterprise Activities shall be described by means of an assignment attribute in the requirements definition and later phases.

6.17.3 Usage

The construct Operational Role is used in requirements definition and later phases. Its template shall enable

a) the capturing of all operational task-related information relevant for the execution of operational processes, and

b) its presentation in a form that is understandable by both humans and machines.

The template shall identify all Enterprise Activities for which this Operational Role is required.

6.17.4 Construct template for Operational Role

Header	
Construct label	OPR
Identifier	<model-unique string>
Name	[<adjective> <noun> | <noun>], name of Operational Role, where <adjective> optionally qualifies the Operational Role and <noun> relates to the scope of the Operational Role
Design Authority	[[<identifier> "/" <name>] [NIL | ":" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having the authority to design or maintain this particular instance
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
	Not applicable
A2.2 applicable at requirements definition and later phases	
Role Job Description	textual description of operational tasks
Operational Skills	[<skill>]+ instances of which are needed for the operational tasks
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.1 applicable at concept definition and later phases	
	Not applicable
B2.2 applicable at requirements definition and later phases	
Specialization_of	[<identifier> "/" <name>]• of Operational Roles that are a generalization of this particular instance. *The name of the Operational Role instance being defined cannot be used in the list*
B2.3 applicable at design specification and later phases	
Assignments	
Assigned to Organization Unit	[<identifier> "/" <name>] of the Organizational Unit that has responsibility and authority on this Operational Role
Assigned to Enterprise Activities	[<identifier> "/" <name>]+ instances of Enterprise Activities to which this role has been assigned

7 Compliance principles

In order to comply with this International Standard, any particular modelling language shall either use the constructs as defined herein and only those constructs or shall be able to provide a mapping for each of its own language constructs to all relevant constructs as defined in this International Standard. This mapping shall include names, mandatory attributes and evidence of the same relationships with other used constructs.

To support models of limited scope, a modelling language can also claim qualified (partial) compliance, by using a subset of constructs as defined herein, or by demonstrating a mapping to these. In this case, the constructs of this International Standard that are not used and conversely the constructs of the modelling language that are not mapped shall both be clearly identified.

A model can also claim compliance with this International Standard if it is

a) a valid construction of a modelling language that is itself compliant, or

b) for a modelling language claiming qualified compliance, if the model uses only those modelling language constructs that can be mapped onto the constructs of this International Standard.

NOTE Two kinds of testing are desired for compliance testing: the construct level specified in this International Standard and the model execution level. In the latter case, the emphasis is on compliance tests using formal description techniques to describe the system behaviour, whereas in the former case the emphasis is on conformance tests.

Annex A
(normative)

Behavioural rules — Detailed description and syntax

A.1 Textual description

A.1.1 Overview

This clause describes the rules introduced in 6.3.5 in textual form with examples. The formal syntax for behavioural rules is contained in A.2. In the following, italicized words or word combinations are denotations explained in text below. Reserved words are denoted totally in upper case[35)].

Behavioural rules are defined with the following general form:

WHEN *condition* DO *action*;

where:

a) a ";" character terminates a behavioural rule;

b) *condition* is one or more of the following three:

1) the occurrence of one or more designated events: as further described in A.1.2.1, these include normal events represented by Event instances, START events for Business Processes, and exceptions;

2) the completion of one or more Business Processes or one or more Enterprise Activities (each of the latter having an ending status[36)], or some combination of both, that have been previously initiated as a consequence of an action of the behavioural rules of the Business Process that contains these rules (see A.1.2.2);

3) a multi-part condition, consisting of either:

i) two or more conditions separated by the reserved word AND, in which case the multi-part condition is defined to occur if all of those conditions occur, or

ii) two or more conditions separated by the reserved word OR, in which case the multi-part condition is defined to occur if any of those conditions occur.

In the case where both AND and OR separators are used in one multi-part condition, parentheses "(" and ")" or equivalent grouping symbols shall be used to group conditions so as to resolve ambiguities.

EXAMPLE The condition START AND (EV1 OR EV2) occurs if START and either EV1 or EV2 occurs, while (START AND EV1) OR EV2 occurs if START and EV1 occur or EV2 occurs.

35) The reserved keywords occurring in this annex are AND, ANY, ASYNCHRONOUS, DO, ES, EXCEPTION, FINISH, FROM, GENERATE, OR, REPEAT, RUN-TIME CHOICE, SELECT, START, SYNCHRONOUS, TIMES, UNTIL, UNORDERED, WHEN and XOR. They can be replaced by any equivalent term provided these are understandable to the human reader.

36) A Business Process does not have an ending status.

c) *action* is one or more of the following:

1) initiate one or more Business Processes or Enterprise Activities or both (A.1.3.1) [37)];

2) complete a Business Process with or without raising an exception (A.1.3.2);

3) generate an Event (A.1.3.3)[38)];

4) repeat a particular action (looping) (A.1.3.4);

5) a *multi-part action* as described below.

A multi-part action is represented as a list of actions separated by commas, which, if more than one Business Process or Enterprise Activity is to be initiated, shall be preceded by a reserved word (indicating either an *and modality* [see A.1.3.1 b)] or an *exclusive or modality* [see A.1.3.1 c)]) denoting constraints on their otherwise parallel initiation (see A.1.3.1). Commas may be replaced by the (semantically equivalent) reserved words AND (for an *and modality*) or XOR (for an *exclusive or modality*).

In this annex, Business Processes and Enterprise Activities are collectively denoted by BPEA, with instances BPEA1, BPEA2, etc.

NOTE 1 Since Behavioural rules are embedded within a Business Process as the behavioural rule set attribute, they do not have an associated identifier.

NOTE 2 The two kinds of condition and four kinds of action, together with their various qualifying modalities, can be used in certain combinations to express various patterns of sequencing relationships between Business Processes or Enterprise Activities, or some combination of both (synchronous/asynchronous; fan-out/fan-in, etc.). Some example patterns, and the corresponding conditions and actions for these, are summarized in Table A.1. Other kinds of patterns, conditions or actions might exist or be developed later.

37) The syntax does not of itself prevent infinite loops, as in a Business Process BP-A that has WHEN START DO BP-A as a behavioural rule, or more indirectly WHEN START DO BP-B while BP-B (or another Business Process invoked directly or indirectly by BP-B) also contains a rule WHEN START DO BP-A. Implementations may impose restrictions, e.g. that no Business Process can initiate itself or a Business Process that initiates this containing Business Process, either directly or indirectly, but such restrictions are implementation matters outside the scope of this International Standard.

38) As with initiation of a Business Process, implementations may impose restrictions in order to avoid infinite loops, e.g. that no Business Process should respond (via a condition in a behavioural rule) to an Event that it generated, either directly or indirectly, but such restrictions are implementation matters outside the scope of this International Standard.

Table A.1 — Example behavioural rule patterns with corresponding conditions and actions

Typical patterns (non-normative)	Condition	Possible actions (and modalities) [see A.1.3]
Start-up process initiation – the unconditional initiation of one or more BPEAs on initiation of this Business Process	START (see A.1.2.1 b)	Initiate action
Event initiation – the initiation of one or more BPEAs following the occurrence of a designated event	Event (see A.1.2.1 a)	Initiate action
Conditional (causing the conditional initiation of a successor BPEA, dependent on the ending status of its predecessor)	Action status-completion (see A.1.2.2 a)	Initiate action
Forced (causing the initiation of a successor BPEA regardless of the ending status of the predecessor)	Action on-completion (see A.1.2.2 b)	Initiate action
Parallel spawning (causing the concurrent, parallel initiation of two or more successor BPEAs)	Any condition	Initiate action: AND-branching (synchronous, asynchronous)
Unordered set (where a set of BPEAs must be executed in total, but the order of execution is not known beforehand and must be determined at run-time)	Any condition	Initiate action: AND-branching (unordered)
Rendezvous (synchronizing the end of spawned BPEAs)	Action AND-completion (see A.1.2.2 c)	Any action
Run-time choice (making an exclusive choice among several possible successor BPEAs)	Any condition	Initiate action: XOR-branching (select, run-time choice)
Alternate completion, on completion of any one of a group of Business Processes or Enterprise Activities	Action OR-completion (see A.1.2.2 d)	Any action except repeat iteratively
Loop (repeating the initiation of a successor BPEA until a given condition holds or for a defined number of iterations)	Action loop completion: REPEAT (UNTIL, TIMES)	Initiate action, loop
Business Process completion (indicating the end of execution of a set of behavioural rules and the termination of its containing Business Process)	Action completion: with ending status	Complete/terminate Business Process: FINISH

A.1.2 Conditions

A.1.2.1 Event

An *event condition* is one of two kinds of conditions used in behavioural rules. The occurrence of a designated event instance is represented by the event name or identifier. Event conditions have three forms:

a) ***normal events*** generated during the normal operation of a Business Process or Enterprise Activity and represented by an Event construct.

b) ***start***, the fact of initiation of this Business Process, denoted by the reserved word START. This is used for example to initiate set-up and preparatory actions within each Business Process. If a behavioural rule set contains more than one behavioural rule, the rules containing START conditions are evaluated first in the same order as they are defined, followed by all other rules, again in the same order as they are defined.

NOTE START can be combined with one or more normal events in a behavioural rule, in which case the behavioural rule will have no effect (no preparatory actions will occur) unless both START and all the designated events have occurred.

c) ***exceptions*** raised as the result of a termination (abnormal ending) of a previously initiated Business Process or Enterprise Activity, or by a special external mechanism such as a watchdog timer, denoted by name of the exception. Exceptions have two forms:

   1) ***designated***, if an exception is raised of the designated kind *exception name* and represented by EXCEPTION *exception name*. The *exception name* shall be unique within the scope in which it is defined and used.

NOTE The reserved word EXCEPTION is optional. It is intended as an aid to the human reader.

   2) ***default*****,** if an exception of any kind is raised that has not been otherwise designated (intended as a catch-all) and represented by EXCEPTION ANY.

A.1.2.2 Action completion

An *action completion* is one of two kinds of conditions used in behavioural rules. Its occurrence signifies the completion of one or more designated Business Processes or Enterprise Activities (each of the latter having a corresponding ending status), or both. When a Business Process or Enterprise Activity has been initiated several times within a loop action as described in A.1.3.4, only the last termination is to be considered as an action completion. Action completion has four forms:

a) ***on-status*** condition – the action is initiated dependent on a comparison of the ending status of an Enterprise Activity with a designated value and represented by one of

   ES(EAx) = status value

   ES(EAx) < status value

   ES(EAx) > status value

   ES(EAx) <> status value

   where EAx is a designated Enterprise Activity, ES(EAx) denotes the ending status of that EAx, and status value is any constant or expression that can be evaluated and tested for equality, less than, greater than or inequality relative to the ending status.

b) ***on-completion*** condition – the action is initiated when a Business Process or Enterprise Activity complete (for the latter, regardless of its ending status provided there is no on-status comparison that has evaluate to true), represented by

   FINISH(BPx) or ES(EAx)=ANY

c) ***AND-completion*** condition (rendezvous) – all of a list of designated Business Processes or Enterprise Activities, or both, complete, represented by

   completion condition BPEA1 AND *completion condition* BPEA2... AND *completion condition* BPEAn

   where each *completion condition* BPEAi is either an on-status condition or an on-completion condition as defined in A.1.2.2 a) and b) above.

d) ***OR-completion*** condition – at least one of a list of designated Business Processes or Enterprise Activities, or both, complete, represented by

   completion condition BPEA1 OR *completion condition* BPEA2... OR *completion condition* BPEAn

   where each *completion condition* BPEAi is either an *on-status* condition or an *on-completion* condition as defined in A.1.2.2 a) and b) above.

A.1.3 Actions

A.1.3.1 Initiate action

The *initiate action* signifies that one or more Business Processes or Enterprise Activities, or both, are to be initiated by a condition occurrence as defined in A.1.2. The initiate action models the fact that a Business Process or Enterprise Activity can be initiated only on completion of its predecessor (see A.1.2.2) or on the occurrence of a start event [see A.1.2.1 b) only for Business Processes], or the occurrence of a normal event [see A.1.2.1 a)], or a raised exception [see A.1.2.1 c)].

EXAMPLE The initiation of an Enterprise Activity can be dependent on the availability of a necessary Resource or Event, even though its predecessor has terminated and enabled it to proceed.

The *initiate action* has three forms:

a) ***single action*** – a single designated Business Process or Enterprise Activity is initiated, represented simply by its name, as in the behavioural rule

   WHEN START DO prepare_Workstation ;

b) ***AND-branching*** – enabling the activation of multiple successor Business Processes or Enterprise Activities, or both, by one or more predecessors. The ordering of successor activation is determined by its *and modality*, which may be synchronous (all starting at the same time), asynchronous (activated in parallel but possibly starting at different times) or unordered (the sequence of activation is not known and will be determined at run-time, possibly subject to temporal constraints). The action for AND process-branching is represented by

   and modality BPEA1, BPEA2,..., BPEAn

   where *and modality* is one of

   SYNCHRONOUS

   ASYNCHRONOUS

   UNORDERED

   The commas in the list of BPEAs may be replaced by the reserved word AND.

c) ***XOR-branching*** – enabling the activation of one and only one successor Business Process or Enterprise Activity by one (or more) predecessor(s). The selection of successor activation is determined by its *exclusive or modality*, which may be case-determined (indexed, or determined by the run-time value of some variable), or run-time choice (determined by a decision made at run-time). The action for XOR-process-branching is represented by

   exclusive or modality BPEA1, BPEA2,..., BPEAn

   where *exclusive or modality* is one of

   SELECT jth item FROM (used for selection of BPEAi by a run-time value of j)

   RUN-TIME CHOICE (used for selection by a run-time decision)

   The commas in the list of BPEAs may be replaced by the reserved word XOR.

A.1.3.2 Complete action

The *complete action* is enabled by any condition as defined in A.1.2 and terminates the Business Process. It has two forms:

a) ***normal completion*** – the Business Process terminates normally, represented by the keyword FINISH

b) ***exception termination*** – the Business Process raises an exception (usually as the result of a designated or default exception condition) and thereupon terminates. It is represented by

GENERATE exception name AND FINISH ;

A.1.3.3 Generate event action

The *generate event action* enables a Business Process to generate an instance of an Event. It is represented by

GENERATE event name ;

A.1.3.4 Loop action

The *loop action* signifies the repetitive initiation of some action. A *loop action* has two forms:

a) ***repeat conditionally*** – a test condition is evaluated after each repetition of the action to be repeated. If the condition is true the loop exits, otherwise the action is repeated and the condition is checked again. Conditional looping may be represented by

REPEAT *action* UNTIL *test condition* ;

where *action* is either initiate action (single action, AND-branching or XOR-branching) or a further loop action, and *test condition* is an action completion as defined in A.1.2.2.

NOTE "REPEAT EAx UNTIL ES (EAx) = ANY" is semantically equivalent to "EAx".

b) ***repeat iteratively*** – providing for a fixed number of repetitions of an action and represented by

REPEAT *action loop count* TIMES ;

where *action* is either initiate action (single action, AND-branching or XOR-branching) or a further loop action, and *loop count* is an expression, which either evaluates to a positive integer that then determines the number of repetitions or, if it does not so evaluate, the action is not initiated.

A.2 Formal syntax in eBNF

The following syntax uses the notation of eBNF as described in ISO/IEC 14977. The normal character representing each operator of *Extended BNF* and its implied precedence is (highest precedence at the top):

* repetition-symbol

\- except-symbol

, concatenate-symbol

| definition-separator-symbol

= defining-symbol

; terminator-symbol

The normal precedence is over-ridden by the following bracket pairs:

‘	first-quote-symbol	first-quote-symbol	’
“	second-quote-symbol	second-quote-symbol	”
(*	start-comment-symbol	end-comment-symbol	*)
(	start-group-symbol	end-group-symbol	)
[	start-option-symbol	end-option-symbol	]
{	start-repeat-symbol	end-repeat-symbol	}
?	special-sequence-symbol	special-sequence-symbol	?

NOTE A space character enclosed in quotes as in “ ” denotes that a literal space character is required, otherwise space characters and line endings (so-called white space) have no significance. A meta identifier can occur on both the left and right sides of a rule, so enabling recursion.

EXAMPLE list = item | item AND list ;

(* uses recursion to generate a non-empty sequence of item, item AND item, item AND item AND item, etc. *)

The following base declarations define positive integers and certain behavioural rule strings:

non zero integer = ‘1’ | ‘2’ | ‘3’ | ‘4’ | ‘5’ | ‘6’ | ‘7’ | ‘8’ | ‘9’ ;
decimal digit = ‘0’ | non zero integer ;
positive integer = non zero integer {decimal digit} ; (* used in XOR-modality and loop *)
upper case letter = ‘A’ | ‘B’ | ‘C’ | ‘D’ | ‘E’ | ‘F’ | ‘G’ | ‘H’ | ‘I’ | ‘J’ | ‘K’ | ‘L’ | ‘M’ | ‘N’ | ‘O’ | ‘P’ | ‘Q’ | ‘R’ | ‘S’ | ‘T’ | ‘U’ | ‘V’ | ‘W’ | ‘X’ | ‘Y’ | ‘Z’ ;
lower case letter = ‘a’ | ‘b’ | ‘c’ | ‘d’ | ‘e’ | ‘f’ | ‘g’ | ‘h’ | ‘i’ | ‘j’ | ‘k’ | ‘l’ | ‘m’ | ‘n’ | ‘o’ | ‘p’ | ‘q’ | ‘r’ | ‘s’ | ‘t’ | ‘u’ | ‘v’ | ‘w’ | ‘x’ | ‘y’ | ‘z’ ;
letter = upper case letter | lower case letter ;
string character = letter | decimal digit | ‘_’ | ‘-‘ | ‘/’ | ‘ ‘ ; (* note that a string character can be a space *)
name = letter, {string character} ; (*A name in a behavioural rule is an identifying label or variable. It must start with a letter, can contain spaces, etc. *)
bp = name ; (* identify Business Process by its name *)
ea = name ; (* identify Enterprise Activity by its name *)
bpea = bp | ea ; (* see A.1.1 *)
ev = name ; (* identify Event construct by its name *)
status constant = name ; (* a literal constant that can be compared with Ending Status *)

(* The following identify reserved words for behavioural rules: *)

(* Reserved words in behavioural rules are delimited by white space or punctuation *)
AND = ? logical and ? ;
ANY = ? equivalent to a wildcard; a condition that compares ANY with any value always occurs ? ;
ASYNCHRONOUS = ? activated in parallel but possibly starting at different times ? ;
DO = ? initiate action if the controlling condition occurs ? ;

ES = ? Ending status of the Enterprise Activity ? ; (* ES(ea) must evaluate to 0-argument predicate *)
EXCEPTION = ? signal abnormal behaviour ? ;
FINISH = ? signal completion of the Business Process? ;
FROM = ? select an identified item from a list ? ;
GENERATE = ? cause an instance of an Event or exception to occur ? ;
OR = ? logical or ? ;
REPEAT = ? iterate ? ;
RUN-TIME CHOICE = ? select an item from a list at the time when a decision is needed ? ;
SELECT = ? identify which list element is to be selected ? ;
START = ? condition that occurs when a Business Process is initiated ? ;
SYNCHRONOUS = ? all starting at the same time ? ;
TIMES = ? identify number of repetitions ? ;
UNORDERED = ? the sequence of activation is not known and will be determined at run-time, possibly subject to temporal constraints ? ;
UNTIL = ? check for action completion ? ;
WHEN = ? wait for the occurrence of a (possibly multi-part) condition? ;
XOR = ? logical exclusive or ? ;
(* The preliminaries end here. *)

(* Define behavioural rule set for Business Process constructs: *)

behavioural rule set = behavioural rule {behavioural rule} ; (* see 6.3.5 *)

behavioural rule = WHEN condition DO action ";" ; (* see A.1.1 *)

condition = condition term
| condition term {AND condition term}
| condition term {OR condition term} ; (* see A1.1 b) 3) *)
condition term = single condition
| (condition) ; (* see A.1.1 b) 3) *)
single condition = event occurrence
| action completion ; (* see A.1.1 b) 1) and A.1.1 b) 2) *)

event occurrence = event
| start
| exception ; (* see A.1.2.1 *)
event = ev ; (* see A.1.2.1 a) ; the named Event needs to be included in the containing Business Process's list of Event Inputs *)
start = START ; (* see A.1.2.1 b) *)
exception = designated
| default ; (* see A.1.2.1 c) *)
designated = [EXCEPTION] exception name ; (* see A.1.2.1 c) 1)*)
exception name = name ;
default = EXCEPTION ANY ; (* see A.1.2.1 c) 2) *)

```
action completion = on completion
        | and completion
        | or completion ;                          (* see A.1.2.2 *)
on completion = on status
        | FINISH ( bp ) ;                          (* see A.1.2.2 and A.1.2.2 b) *)
on status = ES ( ea ), compare, status value ;     (* see A.1.2.2 a) *)
compare = "=" | "<" | ">" | "<>" ;                 (* see A.1.2.2 a) *)
status value = status constant
        | name
        | ANY ;                                    (* see A.1.2.2 a) ; status value is
                                                   a constant or expression that evaluates to a name;
                                                   this grammar reflects the ambiguity of the
                                                   language in this respect – status constant and
                                                   name are different semantics but both are a
                                                   sequence of string characters;
                                                   ES(ea) compare ANY is always true if ea
                                                   is complete *)
and completion = (on completion {AND on completion}) ;   (* see A.1.2.2 c) *)
or completion = (on completion {OR on completion }) ;    (* see A.1.2.2 d) *)

action = initiate
        | complete
        | generate
        | loop ;                                   (* see A.1.3 *)


initiate = single action
        | and branching
        | xor branching ;                          (* see A.1.3.1 *)
single action = bpea ;                             (* see A.1.3.1 a) *)
and branching = and modality, and action set ;      (* see A.1.3.1 b) *)
and modality = SYNCHRONOUS | ASYCHRONOUS | UNORDERED ;
                                                   (* see A.1.3.1 b) *)
and action set = single action {"," single action}
        | single action {AND single action} ;      (* see A.1.3.1 b) *)
xor branching = xor modality, xor action set ;     (* see A.1.3.1 c) *)
xor modality = RUN-TIME CHOICE
        | SELECT nth action FROM ;                 (* see A.1.3.1 c) *)
xor action set = single action {"," single action}
        | single action { XOR single action} ;     (* see A.1.3.1 b) *)
nth action = positive integer ;                    (* see A.1.3.1 c) ; nth action is
                                                   an expression that should evaluate to a positive
                                                   integer <= number of items in the xor action set,
                                                   otherwise an exception is raised *)


complete = [raise exception] normal ;              (* see A.1.3.2 *)
raise exception = GENERATE exception name AND ;    (* see A.1.3.2 b) *)
normal = FINISH ;                                  (* see A.1.3.2 a) *)
```

```
generate = GENERATE event ;                                  (* see A.1.3.3 *)

loop = conditional repeat
       | iterative repeat ;                                  (* see A.1.3.4 *)
conditional repeat = REPEAT action UNTIL action completion ;
                                                             (* see A.1.3.4 a) *)
iterative repeat = REPEAT action, loop count TIMES ;         (* see A.1.3.4 b) *)
loop count = positive integer ;                              (* see A.1.3.4 b) ; loop count should
                                                             be an expression that evaluates to a positive
                                                             integer or zero (in which case the action is not
                                                             executed), or if it does not so evaluate, an
                                                             exception is to be raised *)

                                                             (* Behavioural rule set definitions end here. *)
```

Annex B
(informative)

Rationale

B.1 Background

European standardization in the field of advanced manufacturing technology is being undertaken by CEN/TC 310, *Advanced manufacturing technology*. In turn, its working group CEN/TC 310 WG1, *Systems architecture*, hereafter referred to as TC 310 WG1, is concerned with standardization work in the field of CIM Systems Architecture. This work is a precursor and a contribution to the development of CEN and ISO standards in this area.

In 1990, CEN/CENELEC WG-ARC (the predecessor to TC 310 WG1) completed ENV 40003, which was later revised as ISO 19439. This International Standard sets out a basis for identifying and co-ordinating the common conceptual constructs necessary for the computer-based modelling of enterprises, focusing on discrete parts manufacturing.

In 1992, WG-ARC completed an Evaluation of Constructs for Function View as defined in ENV 40003 and that evaluation has been published by CEN/CENELEC as Technical Report R-IT-06. The evaluation showed that no one initiative contained all the methods, constructs, semantics and representation that were needed and that additional input was required from projects active on this area. Since that time, major new input has been received (see References [11] and [12]). This International Standard incorporates that input but the basis for the work and the terminology used continues to be ISO 19439.

CEN/TC 310 WG1 recognizes that the discipline of enterprise modelling is under continuous development, for example in its extension from application in one commercial company to the extended, virtual enterprise requiring that all process and entities dealt with in the enterprise are modelled as information. For this reason, new concepts and associated constructs are likely to emerge incrementally and replace or modify those described herein, which are therefore to be regarded as a view of today's state of the art.

B.2 The framework for enterprise modelling — ISO 19439

ISO 19439 sets out a number of modelling concepts for CIM requirements and for the various perceptions of the needs of the enterprise modeller. The concepts have been accepted as being sufficiently general to describe a wide range of manufacturing operations, products, systems and services and the various concerns of those responsible for achieving the goals of CIM by deploying suppliers and resources.

From all possible dimensions of modelling, the framework selected three for their ability to represent the concepts that are required.

— One dimension is concerned with the development and evolution of the enterprise model itself, starting from the identification of the enterprise domain to be modelled and the statement of requirements to a processable model and its later refinements; this is the dimension of enterprise model phase.

— One dimension is concerned with the structure and behaviour of a model to express key aspects of an enterprise; this is the dimension of enterprise model view.

— One dimension is concerned with the specialization of models and model components from the general to the particular; this is the dimension of genericity.

B.3 The vision for enterprise modelling

Because of the evolving state of the art, the modelling framework and its related language constructs should not restrict the choice of methods used in the development, operation and evolution of CIM models but their use should assist the alignment of model components with an enterprise's particular mission, objectives, structure and operations.

As different enterprises have both similar and very particular requirements, components are to be aggregated from generic and partial modelling language constructs which have been chosen and specialized to accommodate specific enterprise requirements.

Due to the complexity of the task of realizing CIM, computer-aided methods are required. To allow this, the required specific properties and behaviours should be represented as far as possible in formalisms capable of computer manipulation and in a graphic representation to assist human comprehension.

A modern vision for enterprise modelling consists in viewing the enterprise both as

a) an open set of concurrent Business Processes achieving enterprise objectives and goals, and

b) a federation of Functional Entities (machines, humans or applications) that use Enterprise Objects to execute process steps of Business Processes.

This vision is compliant with the structure of the ISO 19439 where

— the function view represents the Business Processes, their Enterprise Activities and necessary initiating Events,

— the information view represents the Enterprise Objects as identified in the function view,

— the resource view represents the Functional Entities as active Resources, Capabilities and Person Profiles and associated Roles, and

— the organization view places Business Processes, Enterprise Objects and Functional Entities under some responsibility and authority defined as Organization Units and Organization Roles.

B.4 How the constructs support this vision

The constructs provide a common language intended to help the industry build a common perception of enterprise models and a common culture for describing these models, as explained in more detail below.

— The process of enterprise modelling, i.e. the representation of different manufacturing system elements and aspects in integrated, computer-processable models, is to be enabled by generic, easy-to-use, standardized constructs. The generic constructs may be specialized or organized into structures or some combination of both[39)] for a specific purpose, e.g. for an industry sector or for a particular kind of enterprise concern such as maintenance. In turn, such structures and generic constructs are to be used for developing particular models for a particular enterprise.

— The descriptions are to be structured by the dimension of enterprise model view covering different aspects, and the dimension of enterprise model phase covering different levels of detail.

The constructs for enterprise modelling are contained in 3.1 and in Clauses 4, 5 and 6.

39) These are known as partial models in ISO 19439.

B.5 Benefits of using the constructs

The constructs will be used by those business users who are making decisions based on operational rather than technical concerns (i.e. a team or shop floor leader or a user who is also an automation project leader). The constructs will be used for the modelling of business requirements and for the development of complete models for decision support and for model-driven operation control and monitoring. This kind of user needs to be able to understand the constructs during the phases of

— business planning (domain identification, concept definition, requirements definition), to analyse, understand, decide on and represent the business requirements and hence the necessary direction of development,

— design specification, to ensure good communication with modellers, developers and integrators, but also to ensure appropriate reusability of constructs, partial models already existing elsewhere in the company,

— installation and validation (implementation), to make it more reliable and reduce time scales,

— operations (domain operation), when problems occur or changes are needed, to understand the situation more quickly, to improve their communication with developers and integrator,

— decommissioning (decommission definition) of the domain operational system at the end of its life.

A very clear, simple and understandable representation and formalism is therefore needed for constructs to make the communication between business users and model builders easier, more reliable and less ambiguous.

The constructs are also intended to assist a wide community [40)] which includes

a) application users,

b) application architects,

c) communicators,

d) documenters of systems,

e) ICT vendors,

f) model builders (manufacturing systems engineers, consulting engineers, systems integrators),

g) manufacturing technology/ICT strategists,

h) re-engineers,

i) researchers,

j) software toolmakers,

k) standards makers.

40) It is recognized that new skills, training and tools will be needed for the application of model-based integration.

The constructs will help this community by

— assisting the development of modelling methods and tools that can be used by manufacturing personnel rather than ICT specialists,

— improving understanding (of requirements, design and implementation) through common relationships of system elements and a common language, and

— providing support for four key system-modelling qualities (efficiency, understandability, reliability, modifiability).

More specific benefits to be obtained are better and safer decision-making, multiple sourcing of model components, faster model building, the ability to carry out alternative what-if co-operative analyses, the provision of guidance for special out-of-line situations and faster implementation of policy.

Annex C
(informative)

Overview of modelling language constructs and relationships

C.1 Introduction to the metamodel

The purpose of this conceptual model is to provide the reader with an overview of the relationships and attributes of the constructs of Clause 6. The metamodel is formalized using UML Class diagram notations, but it is important to stress that it is not intended as the basis of engineered instantiations or as a starting point for the development of tools. With some exceptions, it does not indicate model phase-related distinctions, nor how they are intended to be used – that information is contained in Clauses 4, 5 and 6.

Because of its complexity, the model that follows is presented in various stages:

a) a diagram that represents the parts, relationships and specializations of modelling language constructs;

b) a diagram providing a high-level view of all 16 constructs;

c) a diagram that represents those constructs and relationships that are involved in the modelling of business functions;

d) a diagram that represents those constructs and relationships that are involved in the modelling of business information;

e) a diagram that represents those constructs and relationships that are involved in the modelling of business resources;

f) a diagram that represents those constructs and relationships that are involved in the modelling of business organization.

To ensure consistency, these diagrams are all generated from a single UML model (class diagram) that encompasses all the constructs and specializations defined in this International Standard. The classes, their attributes and relationships are held in a single data structure, and the diagrams are selective views on that structure – only the layout has been adjusted by hand.

Figures C.3, C.4, C.5 and C.6 are derived from the construct templates of Clause 6 and illustrate the relationships between the concepts involved[41)]. It should be stressed that these diagrams are illustrative – the templates themselves constitute the definitive (and normative) text. They are generated by placing concepts that are central to a view in the middle of the diagram (e.g. Business Process, Enterprise Activity and Functional Entity for function view, Enterprise Object for Information View, etc.). The constructs that have direct association or specialization relationships are then added, as are complementary concepts[42)] (e.g. constraints, integrity rules) and the relationships themselves. Attributes are then added for each construct, unless those attributes correspond to an association, in which case the association is labelled accordingly, or to a generalization or specialization. Associations containing an arrowhead are to be read in the direction of the arrowhead. Labels at each end of some associations indicate the roles played by the corresponding constructs in that association.

41) In contrast, Annex E illustrates model development usages based on an example from ISO 19439.

42) Some associations are inherited via specialization, e.g. in C.4, Product, as a specialization of Enterprise Object, inherits an association with the complementary concept *constraint* and may be represented by Object Views.

NOTE Depending upon the purpose of modelling, constructs are often applicable to several model views and the aspect models presented in items c) to f) above, and the examples in Annex E demonstrate this. For any particular enterprise model, the enterprise modeller will need to decide which constructs appear in which enterprise model view.

These diagrams contain abstract classes (marked as <<abstract>>), which are introduced solely for the purposes of this UML metamodel, and which are not used in other clauses.

C.2 Template for modelling language constructs

Each construct contains a header that is identical in structure for all constructs.

This header part is followed by a body part containing descriptives, which is a construct-specific collection of descriptive attributes, and relationships, which is a construct-specific collection of relational attributes. Some constructs also contain additional or constrained attributes relevant only for a particular enterprise model phase and later phases, or for a particular purpose. Such usages are treated as specializations of the original construct. Clause 6 defines each construct and its attributes using a common template as described in 5.2.

The relationships between these template parts are noted in Figure C.1.

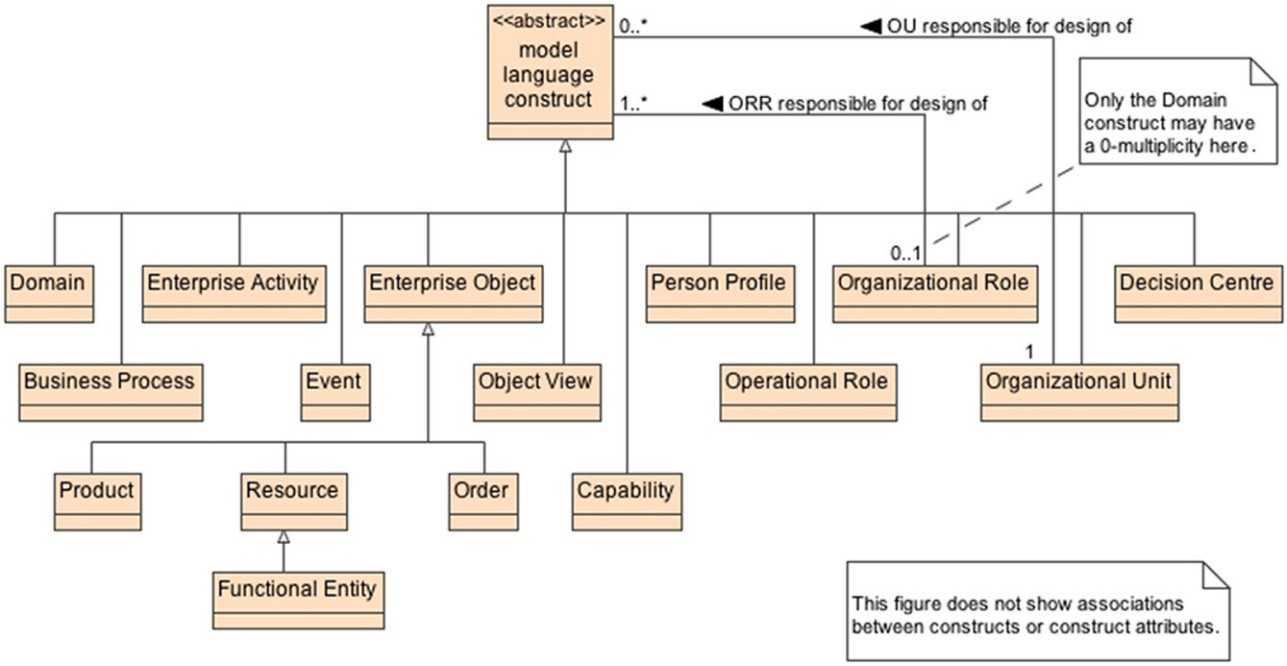

Figure C.1 — Construct template constituents

The complete set of modelling language constructs (without their associations) is noted in Figure C.2 and defined in detail in Clause 6.

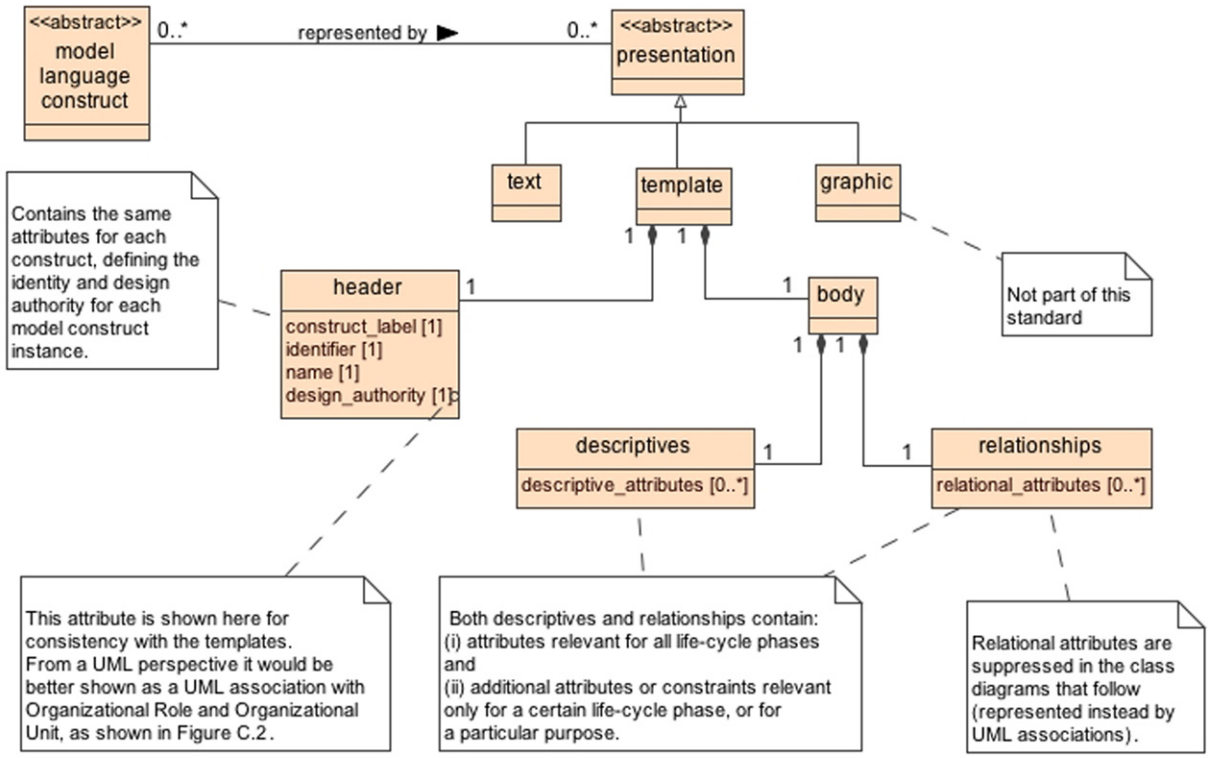

Figure C.2 — The set of constructs (associations not shown)

C.3 Metamodel of functional aspects

As noted in Figure C.3, certain constructs are used to model functional aspects of an enterprise. Behavioural rules are relevant for all phases after requirement definition, only functional entities execute functional operations.

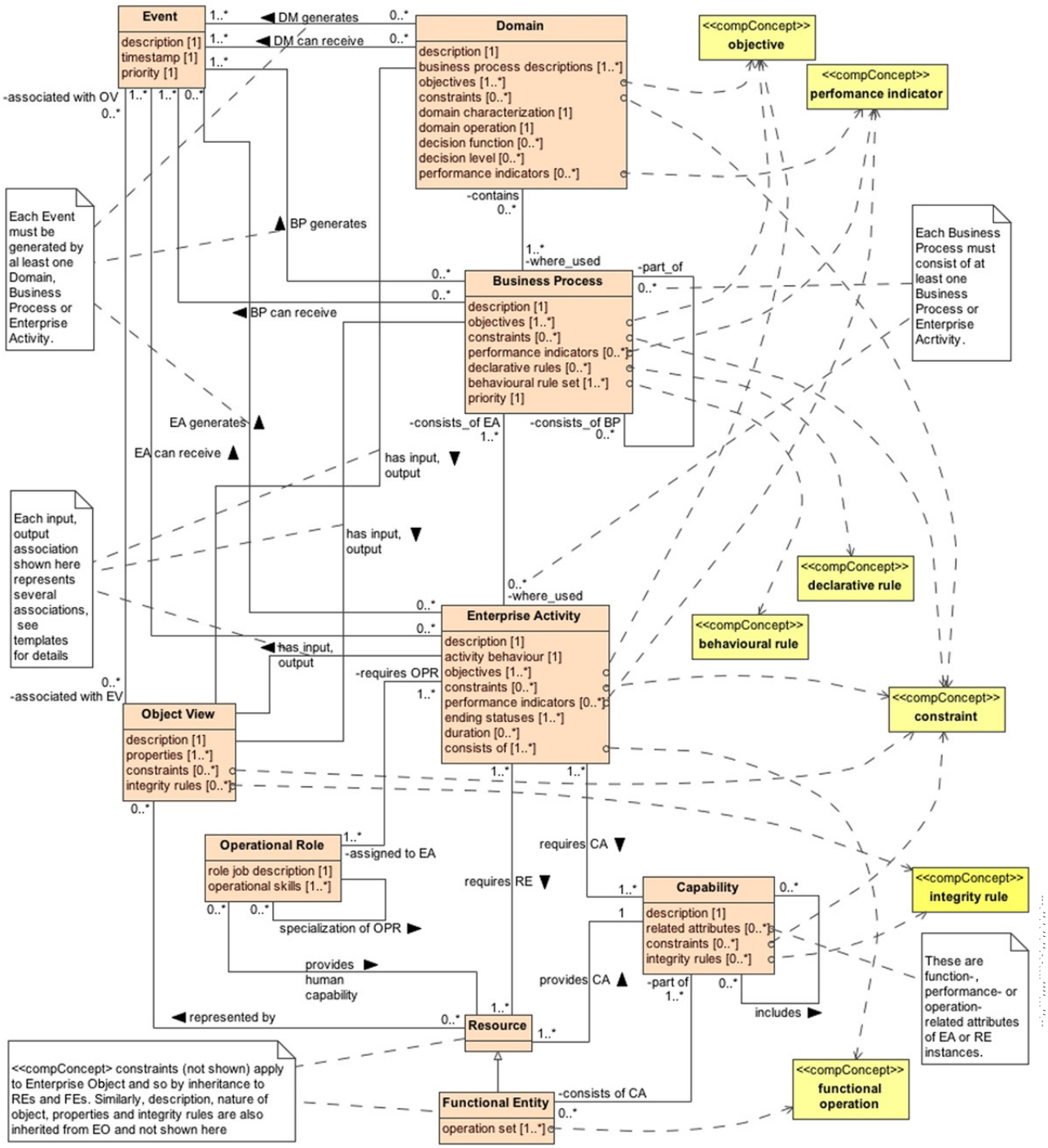

Figure C.3 — Functional aspects

C.4 Metamodel of information aspects with specializations

As noted in Figure C.2, constructs are used to model business information with specializations of Enterprise Object as the constructs Product, Order and Resource. It also illustrates the key role of Object View in presenting a selective view of an Enterprise Object. Object Views are used crucially as inputs and outputs of Domains, Business Processes and Enterprise Activities (see C.3), and to convey optional information associated with some Events.

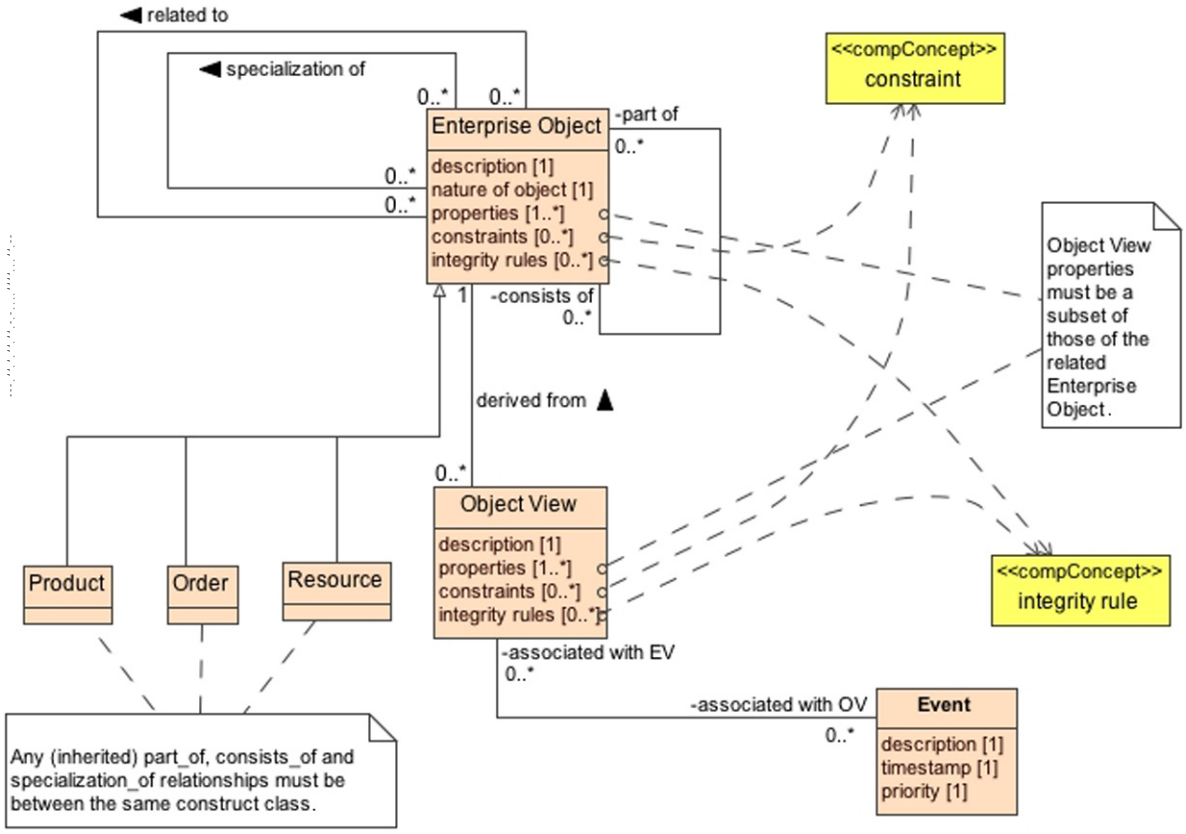

Figure C.4 — Information aspects

C.5 Metamodel of resource aspects

As noted in Figure C.5, different resources provide support for Enterprise Activities and functional operations. It illustrates the CIMOSA modelling paradigm. Person Profile is assigned to Operational Role, which then has a “provides human resource” association with Resources (and by inheritance also Functional Entity) at the requirements definition and later phases.

NOTE Each Functional operation is executed by a Functional Entity.

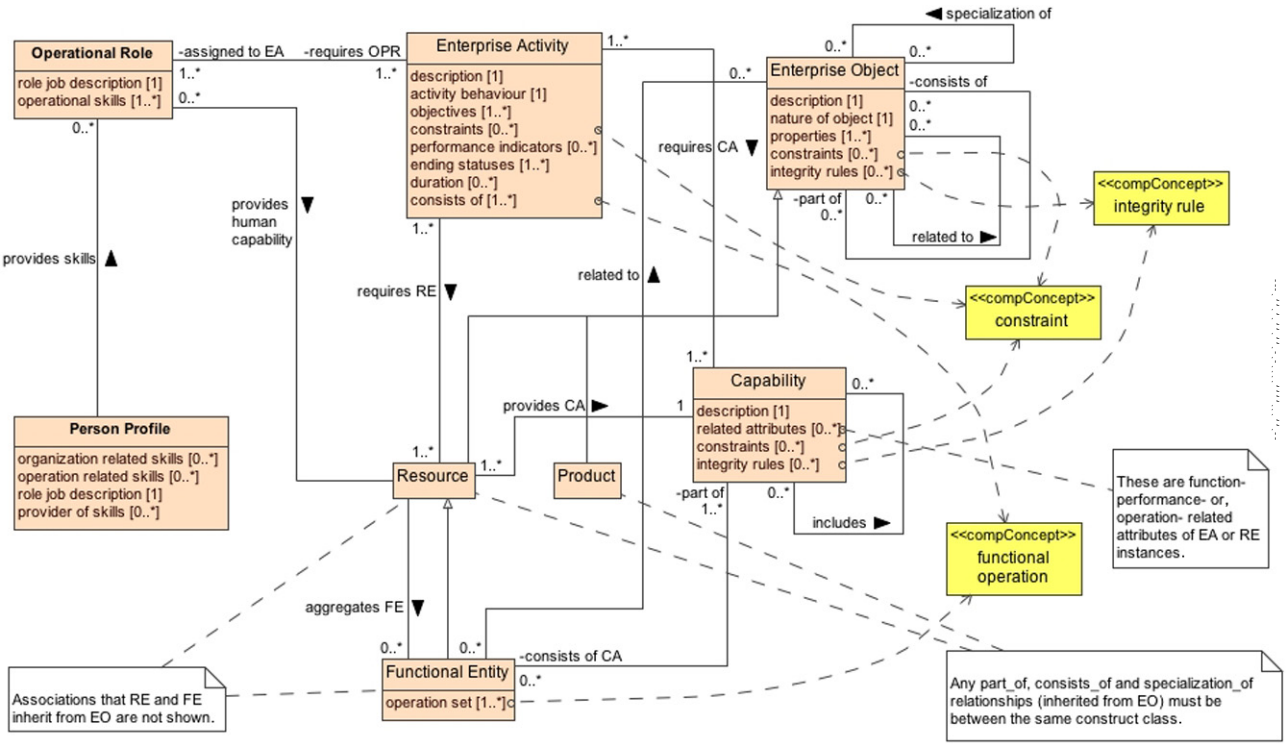

Figure C.5 — Resource aspects

C.6 Metamodel of organizational/decisional aspects

Lastly the various constructs are managed by Organizational Roles and Organizational Units/Decision Centres as noted in Figure C.6. Note that the “OU resp for” association includes an Organizational Unit’s responsibility for other Organizational Units (by inheritance from enterprise model construct) and similarly the “DC decisions apply to” includes a Decision Centre’s decisions applying to other Decision Centres.

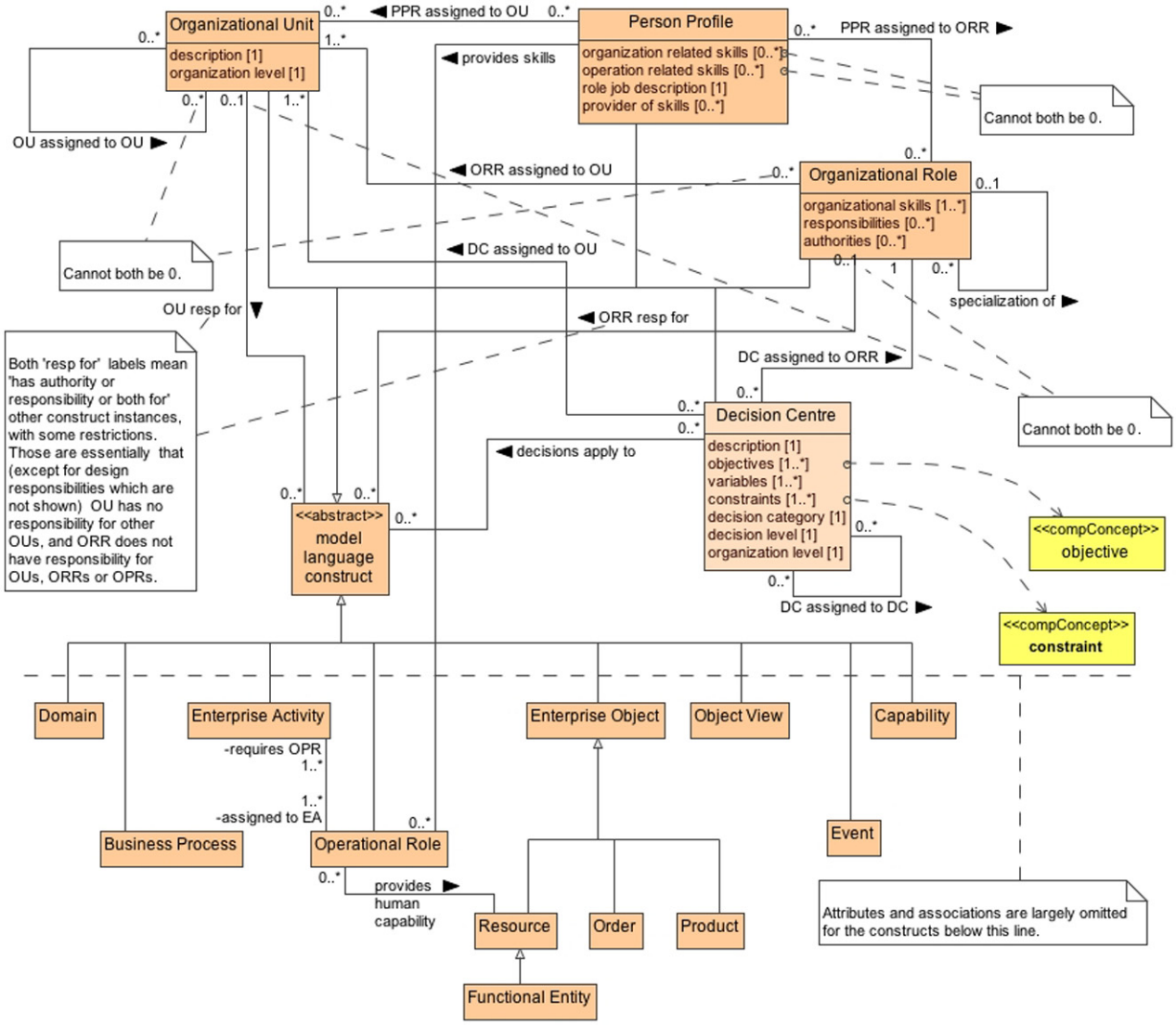

Figure C.6 — Organizational/decisional aspects

Annex D
(informative)

Demonstration of applicability of this International Standard to other initiatives

D.1 Introduction

This annex demonstrates the range of applicability of this International Standard by relating the modelling constructs defined herein to two work items in standardization and in the European research arena respectively. In the first part the comparison is made with the published international standard for the ODP Enterprise Language[4] and in the second part with the first public results of the on-going European ATHENA research project relating to its POP* language[10].

D.2 Metamodel of constructs relative to ISO/IEC 15414

D.2.1 Introduction to ODP and to the comparison

Further to the metamodel presented in Annex C, this clause compares the modelling language constructs described in Clause 6 of this International Standard to the work of ISO/IEC JTC1/SC7/WG19 on the Open Distributed Processing (ODP) Enterprise Language[4]. This work extends the five viewpoints (enterprise, information, computational, engineering and technology) and corresponding languages of ISO/IEC 10746 in order to

— refine and extend the enterprise language defined in ISO/IEC 10746-3 to enable full enterprise viewpoint specification of an ODP system,

— explain the correspondences of an enterprise viewpoint specification of an ODP system to other viewpoint specifications of that system, and

— ensure that the enterprise language when used together with the other viewpoint languages is suitable for the specification of a concrete application architecture to fill a specific business need.

"The ODP Enterprise Language introduces refinements... [of concepts in other standards]... additional viewpoint specific concepts, and prescriptive structuring rules for enterprise viewpoint specifications... [so providing]... a common language (set of terms and structuring rules) to be used in the preparation of an enterprise specification capturing the purpose, scope and policies for an ODP system."[43)]

This clause relates the modelling language constructs presented in this International Standard to the metamodel of the system concepts defined in ISO/IEC 15414 through a mapping of the terms used in the two standards onto each other. The corresponding[44)] terms and definitions in the two standards are listed in the mapping Table D.1. The table also includes some of the terms used in this comparison that are defined in only one of the standards.

---

43) The above objectives and text are taken from ISO/IEC 15414:2006, subclause 0.2.

44) Here and in the rest of this annex, 'corresponding' means some degree of equivalence in the intended application or usage of the term within a somewhat similar context. It does not mean that the terms are semantically interchangeable. In several cases, one term is more or less specialized than the other. Also qualifying notes and explanatory material have been omitted from some definitions.

D.2.2 The ODP Metamodel

Figure D.1 presents the system concepts as identified in ISO/IEC 15414.

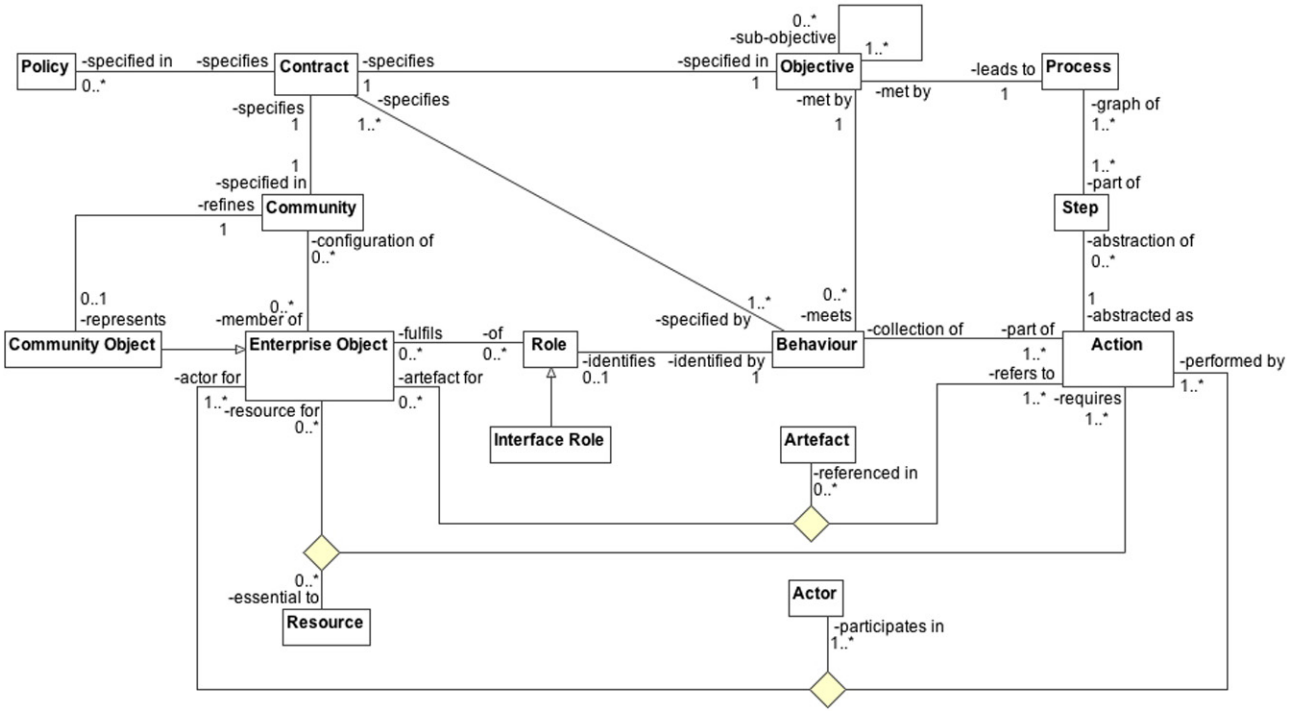

Figure D.1 — Metamodel of ODP Community and Behaviour Concepts

D.2.3 ODP-related metamodel for this International Standard

Figure D.2 is a redrawing of Figure D.1 with some of the ODP system concepts renamed using the corresponding terms used in this International Standard. The usage of the ODP Enterprise Language concepts in this International Standard is indicated in the legend of the figure. Nine of the ODP Enterprise Language concepts are used in the metamodel of modelling language constructs and complementary concepts. Three thereof keep the same term and six are renamed using the related terms defined in this International Standard. Six system concepts are not used directly in the corresponding metamodel for the modelling language constructs of this International Standard, although Role will later be reinterpreted as applicable to persons.

Two of the concepts in Figure D.3 (Objective, Behavioural Rule) are not defined as modelling constructs, but are defined as an attribute of several constructs and as a complementary concept respectively.

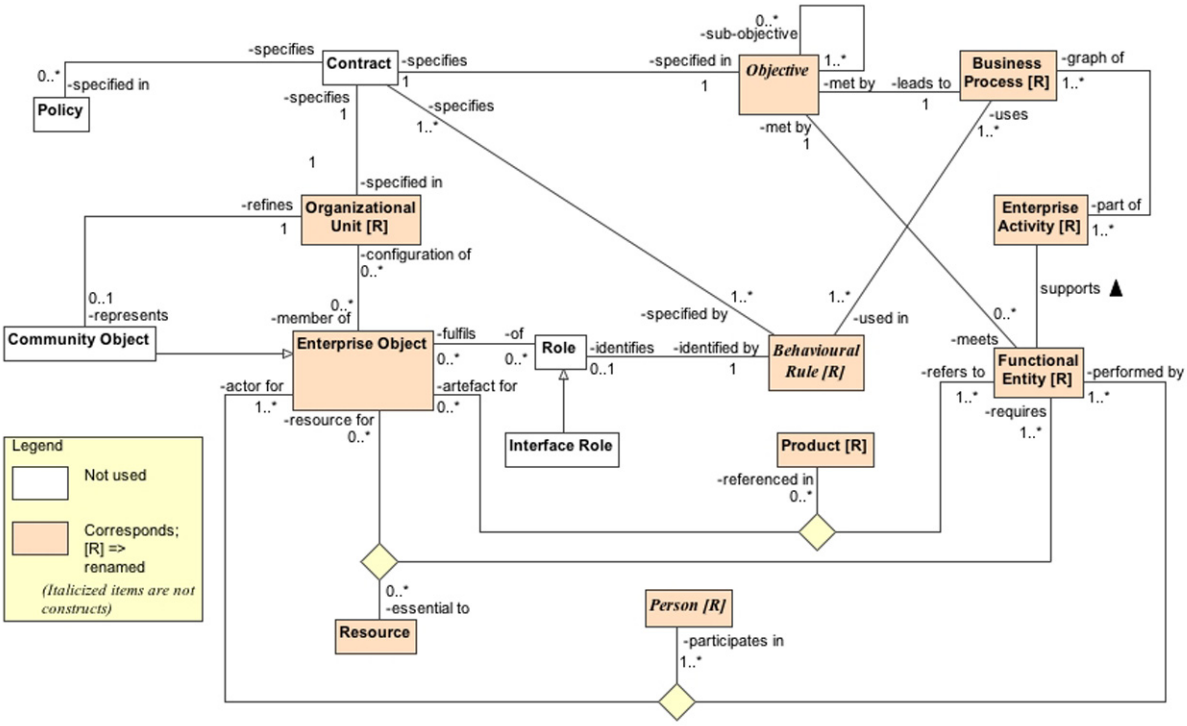

Figure D.2 — Metamodel of ODP Community and Behaviour Concepts modified to fit this International Standard

Nine modelling language constructs are not defined in the ODP Enterprise Language metamodel. One of these (Order) is a specialization of the Enterprise Object construct and is therefore related to ODP concepts. A further three (Person Profile, Organization Role, Operational Role) are specializations of the ODP Role concept. The remaining five constructs (Capability, Decision Centre, Domain, Enterprise Object View and Event) are not contained in the ODP metamodel. However, three of these (Capability, Decision Centre, and Enterprise Object View) can be related to ODP Enterprise Language concepts such as Resource/Actor, Community and Enterprise Object.

Figure D.3 adapts Figure D.2 to correspond more closely to the modelling language construct metamodel of Annex C and shows how Role is reinterpreted to express the association of a Person with an Enterprise Activity and an Organizational Unit. It also recasts Functional Entity as a specialized Resource support for Enterprise Activity.

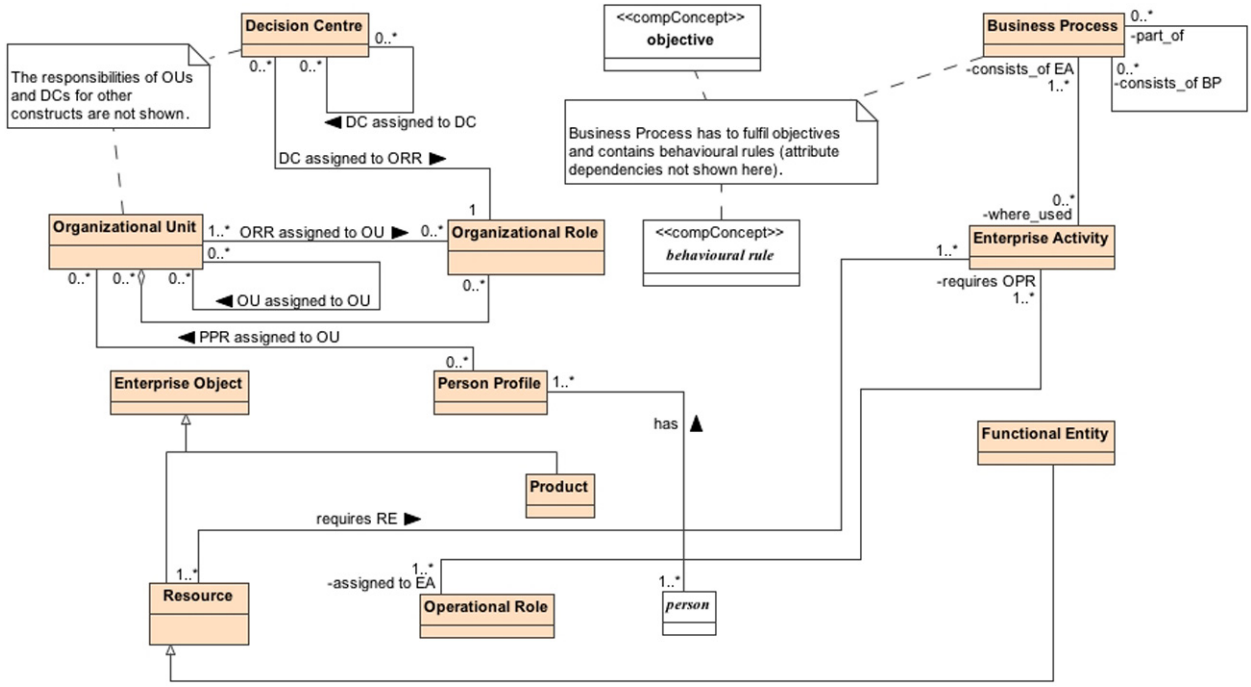

Figure D.3 — Partial metamodel for this International Standard adapted to ISO/IEC 15414

D.2.4 Comparison of definitions

Table D.1 — List of constructs from this International Standard and corresponding concepts from ODP Enterprise Language

ISO/IEC 15414 – ODP Concepts	Constructs from this International Standard	Comments
Action: something which happens (ISO/IEC 10746-2). Every action of interest for modelling purposes is associated with at least one object.	Event: construct that represents a solicited or unsolicited fact indicating a state change in the enterprise or its environment. Functional Entity: construct that is a specialization of Resource construct, which represents an aggregation of Resources able to perform, completely on its own, a (class of) functional operation(s) required by an Enterprise Activity, or part thereof, and to communicate with the related control.	In this International Standard, an Event is a trigger that initiates or is generated by a Business Process or Enterprise Activity. The Functional Entity construct is a specialization of Resource capable of undertaking an action; therefore it would be associated with an ODP action as a resource (actor).
Actor (with respect to an action): A role in which the enterprise object fulfilling the role participates in the action. That object may be called an actor. NOTE – It may be of interest to specify which actor initiates that action.	Person Profile: construct that represents a set of personal capabilities, skills and responsibilities that are required by an Organizational Unit or an Enterprise Activity, or both, and that are provided by a person.	This International Standard specializes the role of an ODP Actor to the Person Profile characteristics of a person (the attributes are those relevant to the requirements of an Organizational Unit or Enterprise Activity).

ISO/IEC 15414 – ODP Concepts	Constructs from this International Standard	Comments
Artefact (with respect to an action): A role in which the enterprise object fulfilling the role is referenced in the action. That object may be called an artefact. NOTE – An enterprise object that is an artefact in one action can be an actor in another action.	Product: construct that is a specialization of Enterprise Object construct, which represents the desired output or by-product of the processes of an enterprise.	As with Person (and Resource), Product in this International Standard corresponds to the enterprise object fulfilling the role, not the role itself.
Capability: not defined.	Capability: construct that represents the collection of capability characteristics (expressed as capability attributes) of either a Resource (its provided Capability) or an Enterprise Activity (its required Capability).	
Community: A community is a configuration of enterprise objects that describes a collection of entities (e.g. human beings, information processing systems, resources of various kinds and collections of these) that is formed to meet an objective. These entities are subject to an agreement governing their collective behaviour.	Organizational Unit: construct that represents an entity of the organizational structure of an enterprise that is described by attributes representing properties of the organization and references to lower level organizational entities.	The ODP community concept models the behaviour of the entity as well as its structure in terms of enterprise objects. The latter would correspond to the set of Organization Units used in this International Standard to represent the enterprise organization.
Decision Centre: not defined.	Decision Centre: construct that represents a set of decision-making activities that are characterized by having the same time horizon and planning period and belonging to the same kind of functional category.	In ODP, a Decision Centre could be modelled as part of the behaviour of a community.
Enterprise Object: An ODP enterprise object is an object in an enterprise specification. Object: A model of an entity. An object is characterized by its behaviour and, dually, by its state. An object is distinct from any other object. An object is encapsulated, i.e. any change in its state can only occur as a result of an internal action or as a result of an interaction with its environment. An object interacts with its environment at its interaction points.	Enterprise Object: construct that represents a piece of information in the enterprise domain that describes a generalized or a real or an abstract entity, which can be conceptualized as being a whole.	This International Standard uses Enterprise Object as a concept to represent information only.
Enterprise Object View: not defined.	Enterprise Object View (Object View): construct that represents a collection of attributes of an Enterprise Object for some particular purpose.	In this International Standard, Object View is a snapshot of relevant attributes — in contrast to ODP's role (of an object in a community) — which encompasses its behaviour.

ISO/IEC 15414 – ODP Concepts	Constructs from this International Standard	Comments
Field of Application (of a specification): the properties that the environment shall have for the specification of that system to be used.	Domain: construct that represents the portion of an enterprise to be modelled providing for identification of the relevant information.	Domain in this International Standard defines both its content and the relation to its environment.
Order: not defined.	Order: construct that is a specialization of Enterprise Object construct, which represents the information for planning and control of Business Processes in an enterprise.	In ODP this information would be modelled as one or more enterprise objects, but the concept is not regarded as a fundamental modelling concept.
Person: not defined. Party: an enterprise object modelling a natural person or any other entity considered to have some of the rights, powers and duties of a natural person.	Person Profile: represents a set of personal capabilities, skills and responsibilities that are (i) required by an Organizational Unit or an Enterprise Activity, or both, and (ii) provided by a person.	Person in this International Standard (and Party in ODP) is a description of an entity in the universe of discourse being modelled rather than a modelling language construct. If modelled directly, it would correspond to the enterprise object fulfilling the role, not the role itself.
Process: a collection of steps taking place in a prescribed manner and leading to an objective. NOTE 1 – A process may have multiple starting points and multiple end points. NOTE 2 – The prescribed manner may be a partially ordered sequence. NOTE 3 – A process specification may be a workflow specification.	Business Process: construct that represents a partially ordered set of Business Processes or Enterprise Activities, or both, that can be executed to realize one or more given objectives of an enterprise or a part of an enterprise to achieve some desired end-result.	In ODP, 'step' is a refinement of Action that corresponds to the Enterprise Activity of this International Standard whereas the ODP 'prescribed manner' corresponds to the rule set that is part of the Business Process of this International Standard. Notes 1 and 2 apply to this International Standard as well, whereas the Process specification of this International Standard would include the workflow specification but would normally be more detailed compared to a workflow specification (Note 3).
Resource: a role in which the enterprise object fulfilling the role is essential to the action and requires allocation or may become unavailable. That object may be called a resource.	Resource: construct that is a specialization of the Enterprise Object construct, which represents the provided capabilities required to execute an Enterprise Activity.	As for Person and Product in this International Standard, the Resource construct does not relate to the ODP concept of role (in an action) but to the enterprise entity fulfilling the role.

ISO/IEC 15414 – ODP Concepts	Constructs from this International Standard	Comments
Role: A role identifies an abstraction of the community behaviour. All of the actions of that role are associated with the same enterprise object in the community. Each action of the community is either part of a single role behaviour or is an interaction that is part of more than one role behaviour. Each of these abstractions is labelled as a role.	Organizational Role: construct that represents a set of organizational capabilities, skills and responsibilities that are required by an Organizational Unit and that are to be provided by a person. Operational Role: construct that represents a set of operational capabilities, skills and responsibilities that are required by an Enterprise Activity and that are to be provided by a person.	See also Person Profile in this International Standard, to which Organizational Role and Operational Role are associated. ODP distinguishes between two types of role: • (community) role, which Role is, and • (action) role – role in an action, which is what actor role, artefact role, resource role are.
Step: an abstraction of an action, used in a process, that may leave unspecified objects that participate in that action.	Enterprise Activity: construct that represents a certain part of enterprise functionality and identifies the Inputs needed for its execution and the outputs created as a result.	In this International Standard, a 'step' exists as Enterprise Activity, which is a part of a process.

D.3 Metamodel of constructs relative to ATHENA – POP* Modelling Language

D.3.1 Introduction to ATHENA and to the comparison

This clause compares the modelling language constructs described in Clause 6 of this International Standard to the work of the ATHENA project: the POP* (Process – Organization – Product – Others) language[10]. ATHENA[45)] (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications) is an Integrated Project sponsored by the European Commission in support of the Strategic Objective "Networked businesses and government" set out in the IST 2003-2004 Work Programme of FP6. Its work package (A1.3) aims at "the development of the core modelling methodology elements for capturing collaborative enterprises design and management" and "tackling the interoperability problem between enterprise models and modelling tools (at higher abstraction level) providing a mapping mechanism".

Reference [10] describes the final version of the ATHENA POP* methodology, providing a set of basic modelling constructs to support model exchange between collaborating enterprises. [The] "work is greatly inspired by already existing initiatives and standards, of which the process oriented BPDM, UEML1.0 and [ISO 19440] are prominent. Although some overlap in motivation and scope may be found between POP* and the initiatives referred to, the ambition of ATHENA is for POP* to take a holistic approach, covering all relevant aspects of (collaborating) enterprises – even though it emphasizes the process oriented aspects. Moreover, POP* is to be a part of a greater whole, providing methodologies and platforms to support modelling, establishing, managing and operating collaborating enterprises."

As in D.2, this clause relates the modelling language constructs presented in this International Standard to the metamodel of the concepts defined for POP* by the ATHENA project and maps the terms used in the two standards onto each other. Rather than one single metamodel as defined in ODP, ATHENA defines for its POP* language a layered meta-architecture. Figure D.4 shows that the POP* modelling dimensions like Process or Product are derived from a POP* Core, which defines common object types based on an Enterprise Knowledge Architecture (EKA) that represents the meta-metamodel.

45) Available at: http://www.athena-ip.org/

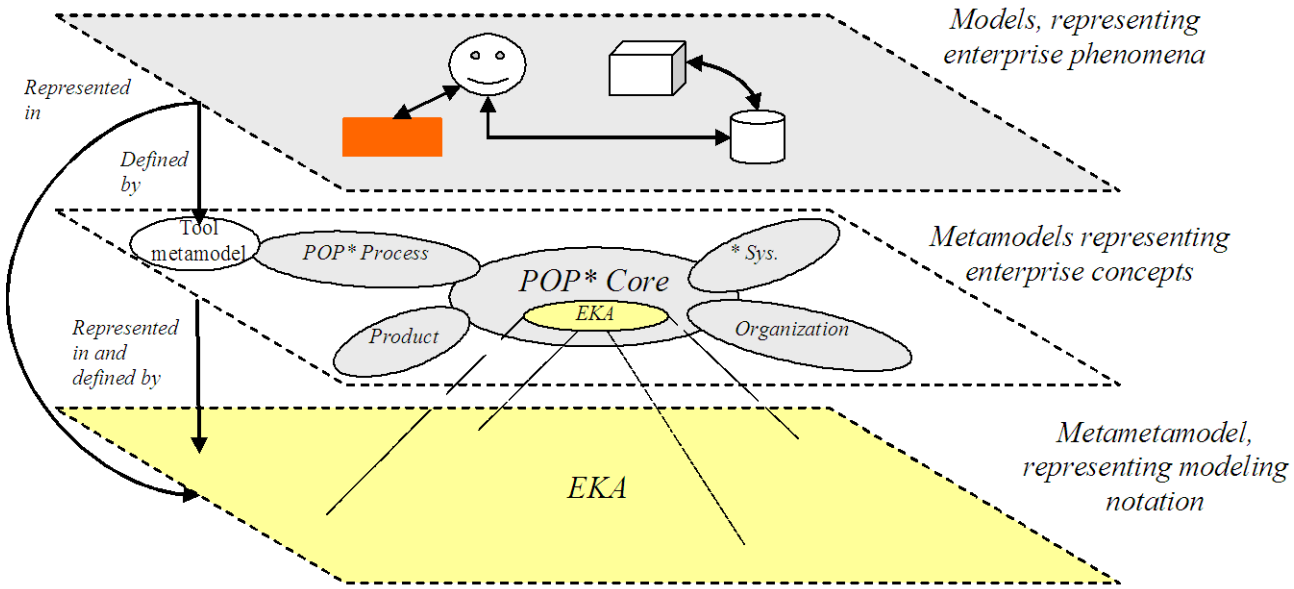

Figure D.4 — ATHENA POP* Architecture

For this comparison, five relevant partial metamodels have been considered (process, product, organization, decision, systems). The corresponding terms defined in the two standards together with their definitions are listed in Table D.3. The list also includes many of the conceptual elements which are defined directly or indirectly in both of the standards.

D.3.2 Metamodels of ATHENA POP*

Figure D.5 shows the process dimension. Stereotypes prefixed by *eka*, such as <<eka Object>> are derived from the POP* Enterprise Knowledge Architecture.

NOTE Figures D.5, D.6, D.7, D.8 and D.9 were generated by the ATHENA project, and the terminology and capitalization reflect project usage (e.g. “Organisation unit” rather than “Organization Unit”).

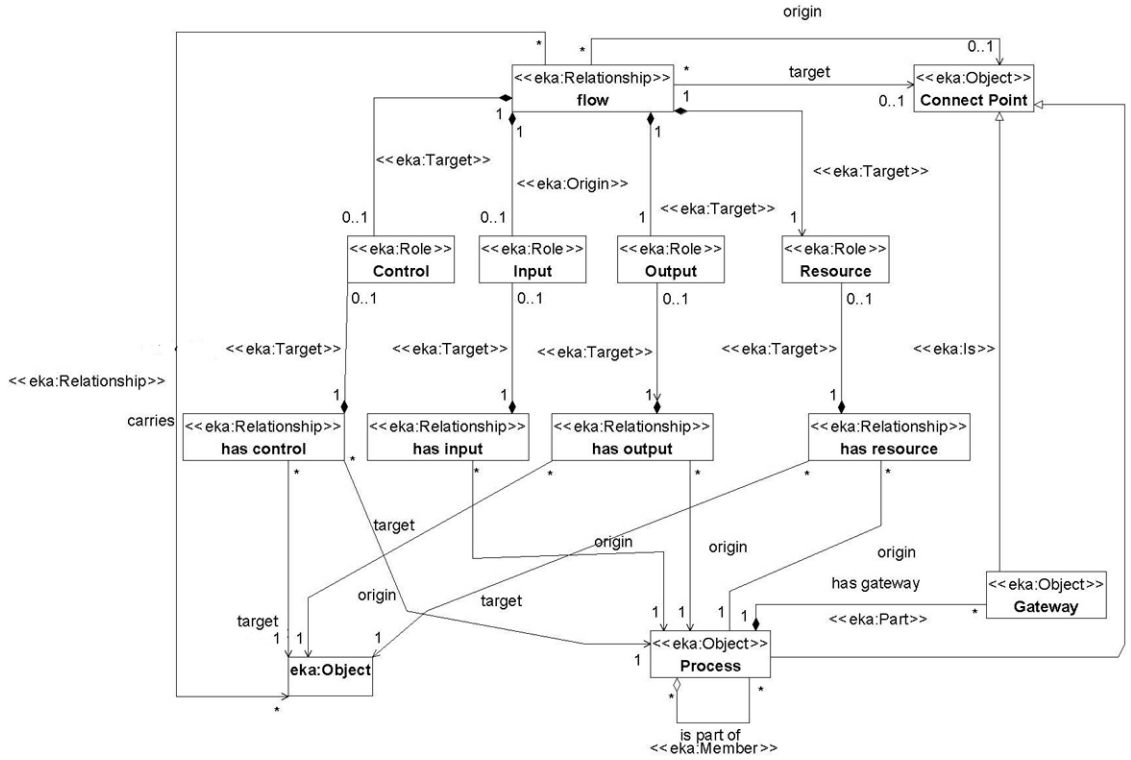

Figure D.5 — POP* metamodel: process dimension

Process is used to represent processes, tasks and activities of any kind. A process can have and be part of a process structure (e.g. work breakdown) through its part of relationship. Processes may be combined into sequences by means of flows and connection points. Processes have attributes such as those given in Table D.2.

Table D.2 — Process attributes

Property name	Value type	Comment
Process Type	{Execution Activity | Decision Activity}	Execution activity: It is a deterministic activity which gives the same result value for the same value of the converging entities (trigger and support). Decision activity: This type of activity can give several result values for the same value of the converging entities (trigger and support) depending on the decision frame and particularly the Decision Variable.
Collaboration Type	{Private Process | View Process | Collaborative Business Process}	Private Processes (PP): PPs refer to a specific organization and are the type of processes that have been generally called workflow processes. View Process (VP): A VP combines one or more PP to an abstract level that enables companies to hide critical information from unauthorized partners. Cross-Organizational Business Processes (CBP): A CBP defines the interactions between two or more business entities. These interactions take place between two or more companies defined as valid sequences of message and/or other material input/output exchanges.
start time	Datetime	The time at which the process starts
finish time	Datetime	The time at which the process is finished
state	Keyword	[Planned, Waiting, Ready, Ongoing, Suspended, Finished, Terminated]

Figures D.6, D.7, D.8 and D.9 illustrate the other dimensions that have been considered in developing the consolidated POP* partial metamodel (see Figure D.10).

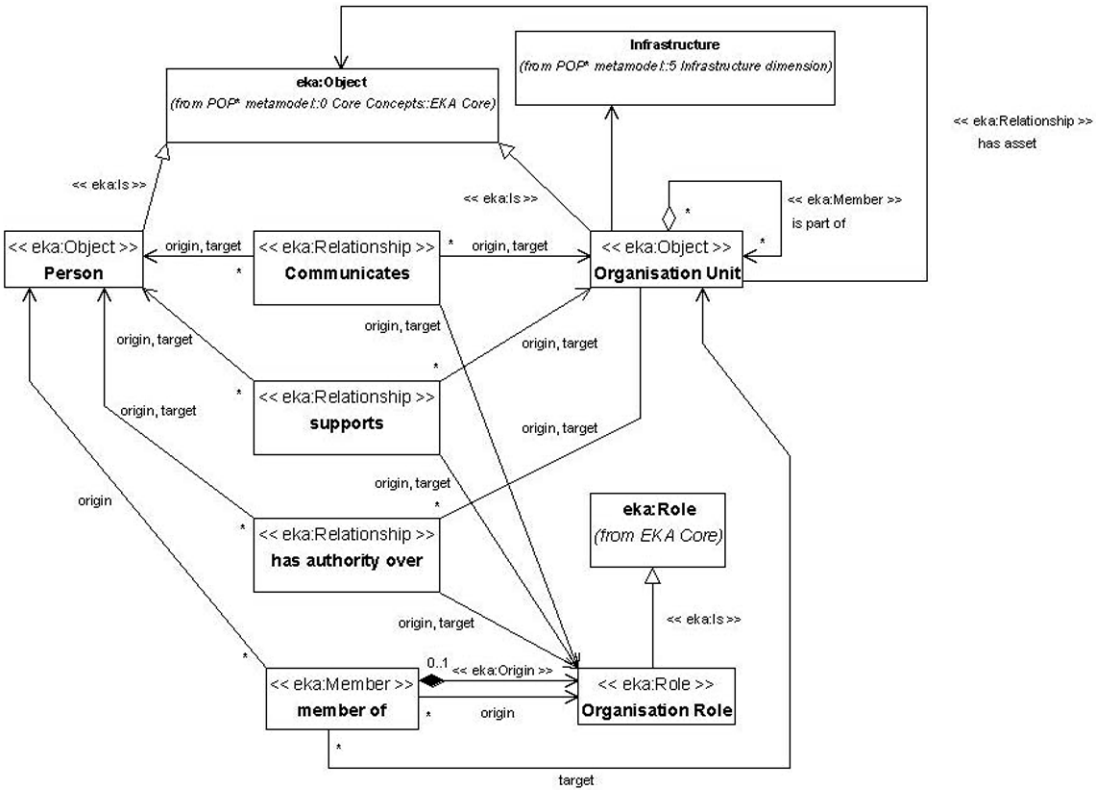

Figure D.6 — The POP* metamodel: organization dimension

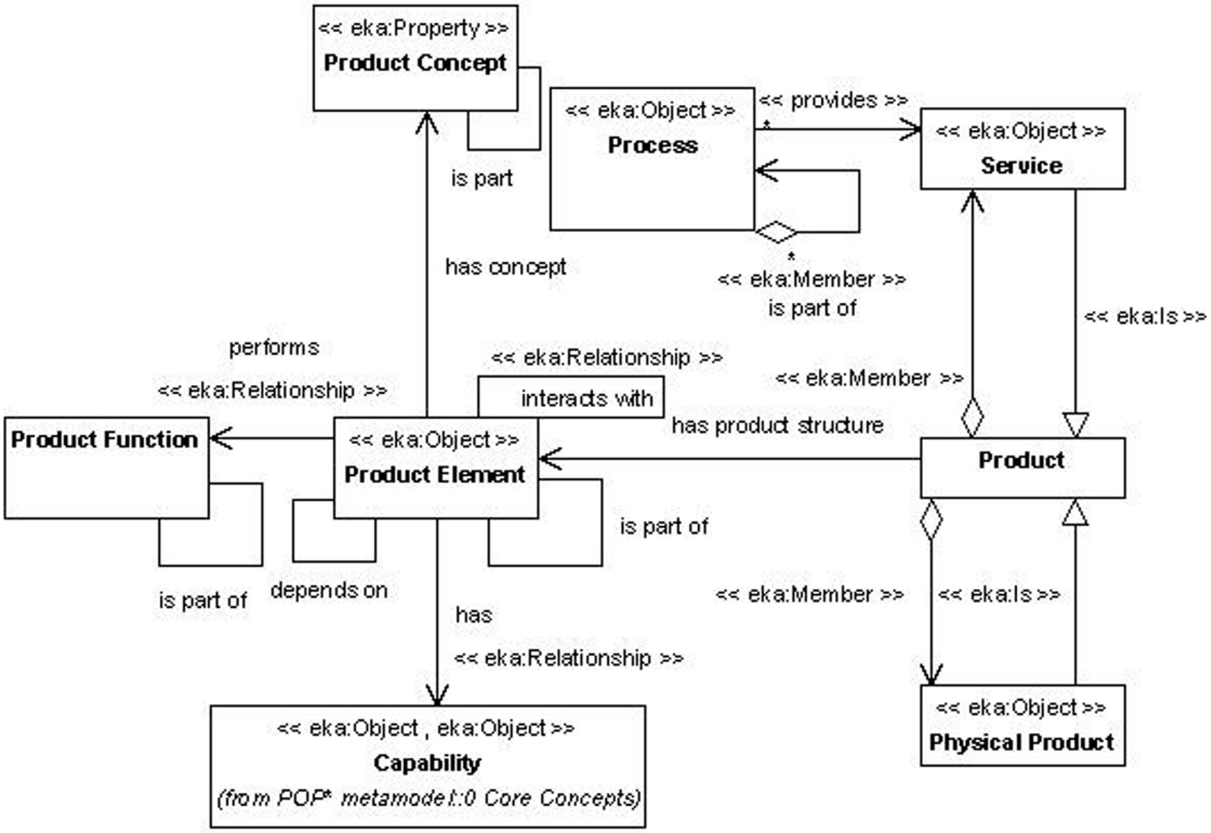

Figure D.7 — The POP* metamodel: product dimension

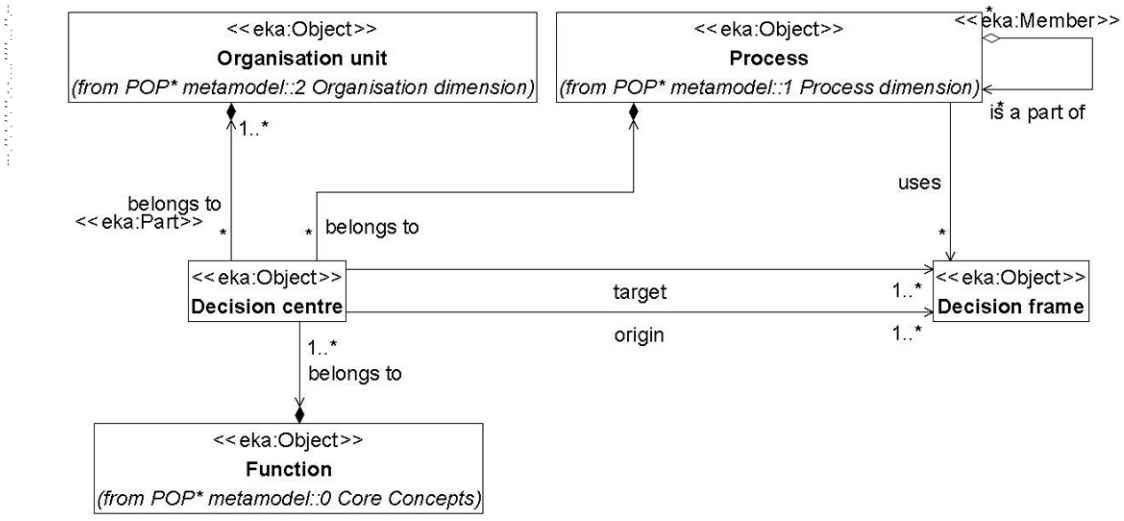

Figure D.8 — The POP* metamodel: decision dimension

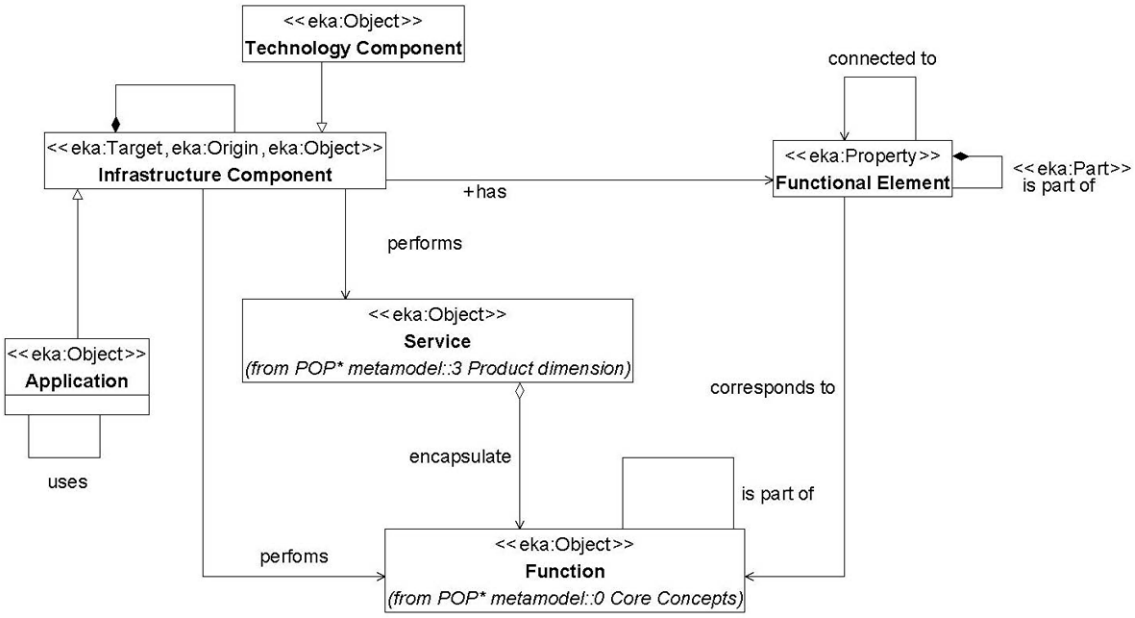

Figure D.9 — The POP* metamodel: system dimension

Figure D.10 presents a consolidated metamodel constructed from the five selected metamodels corresponding to process, product, organization, decision and systems dimensions.

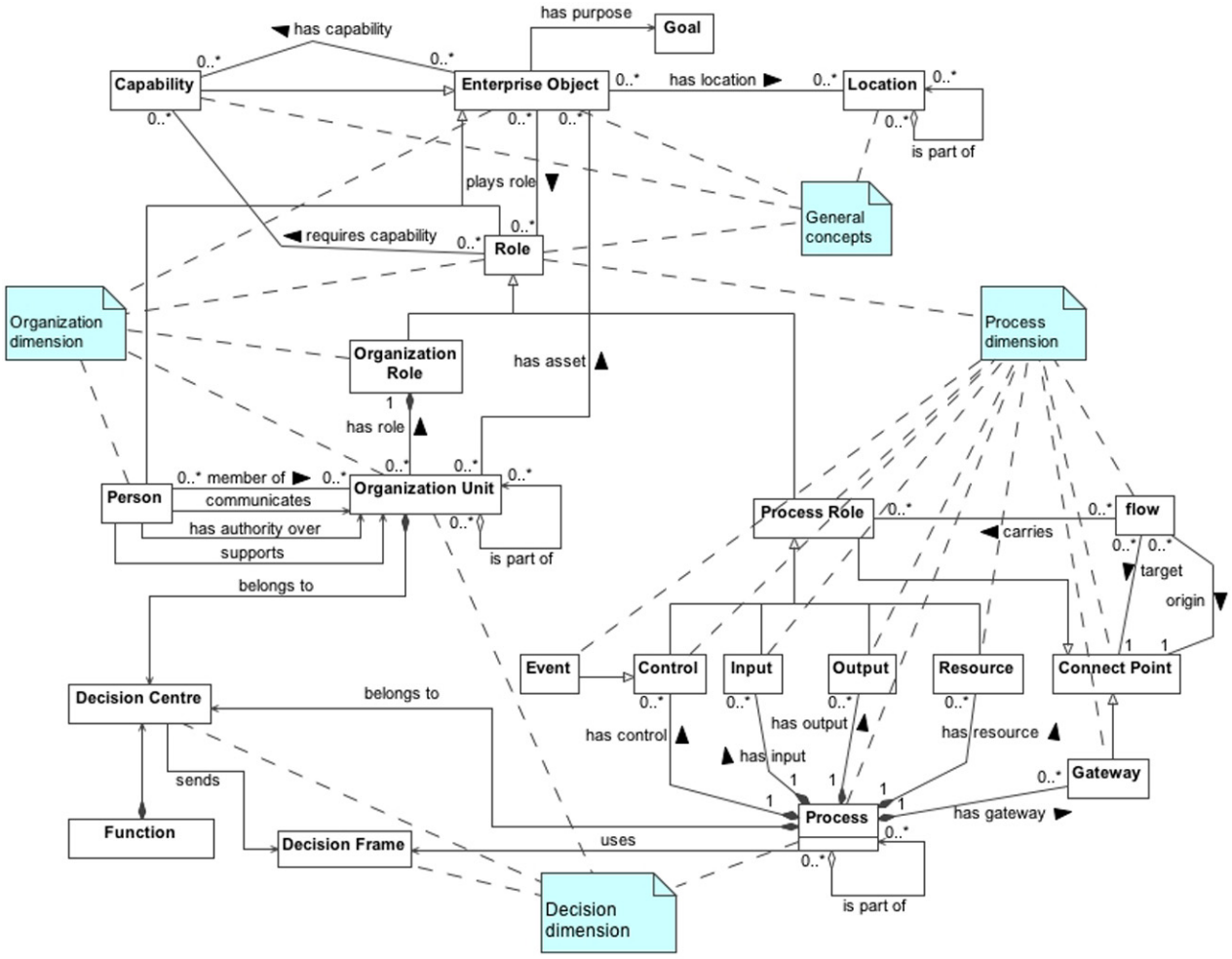

Figure D.10 — Partial metamodel for ATHENA POP*(adapted from four POP* individual metamodels)

D.3.3 POP*-related metamodel for this International Standard

In Figure D.11, Figures D.6 to D10 have been redrawn substituting as far as possible equivalent constructs from this International Standard and renaming where necessary. The usage of POP* concepts in this International Standard is indicated in the legend of this figure. Twelve of the POP* concepts are used in this metamodel, four of these being renamed. Eight POP* concepts are not used in the corresponding metamodel for the modelling language constructs of this International Standard.

Four of the concepts in Figure D.5 (Person, Behavioural Rule, Input, Output) are not defined as modelling constructs, but are defined as basic concept, complementary concept, and attributes of several constructs respectively.

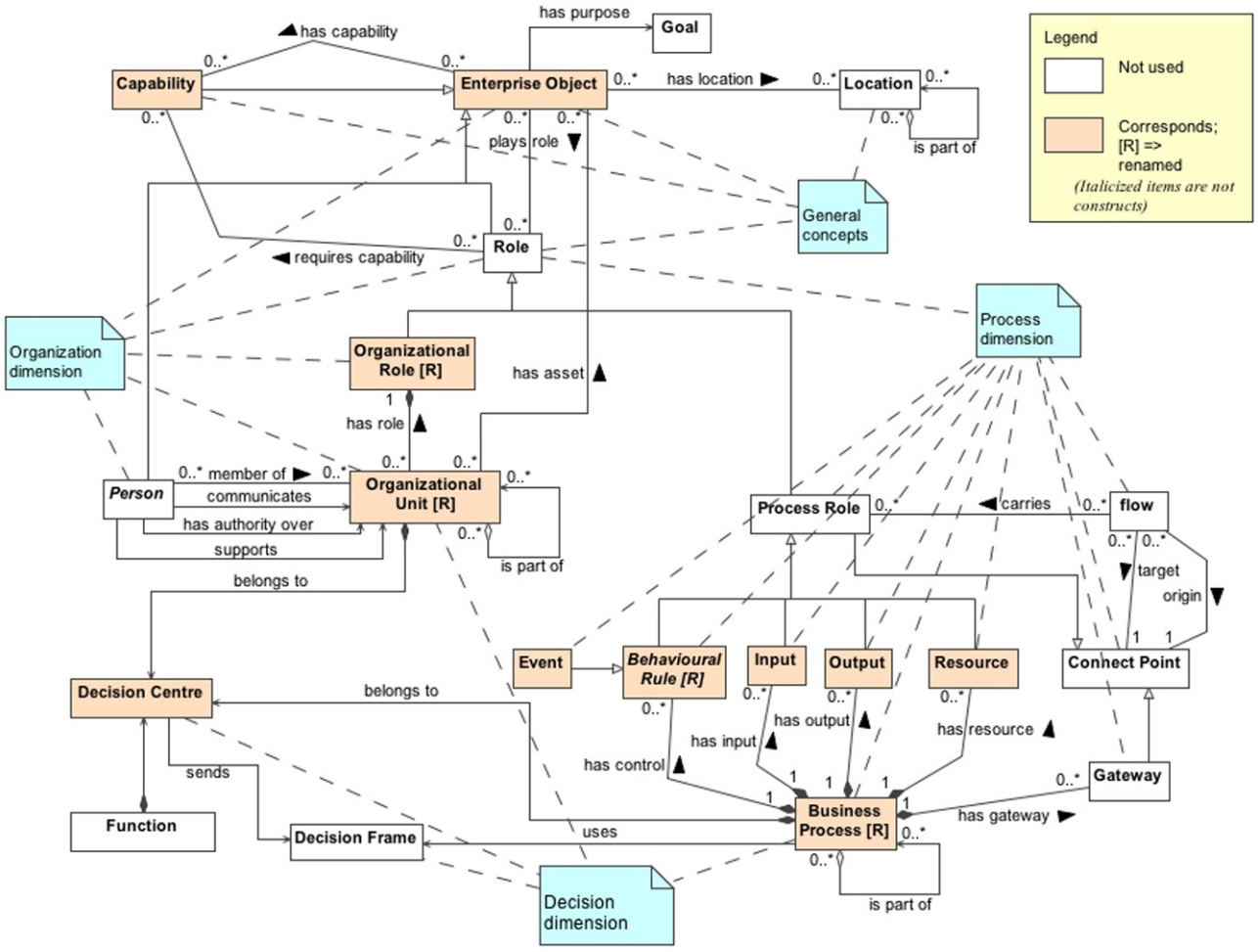

Figure D.11 — Partial metamodel for ATHENA POP* modified to fit this International Standard

Eight modelling language constructs of this International Standard are not defined directly in the partial POP* metamodel illustrated in Figure D.11. Two of these (Product, Order) are kinds of Enterprise Object, while Functional Entity is a specialization of Resource, and Enterprise Activity is a kind of Business Process. The Object View construct is largely equivalent to the inputs and outputs of a POP* process. Domain is a concept that is not used in the selected POP* metamodels. The remaining two constructs (Person Profile, Operational Role) may be regarded as specializations of POP*'s Role and Resource Role.

Figure D.12 adapts Figure D.11 to correspond more closely to the modelling language construct metamodel of this International Standard. It shows in particular how the Operational Role construct represents (human) support for an Enterprise Activity and, in this International Standard, how a person is more closely associated with a Person Profile and hence Organizational and Operational Roles.

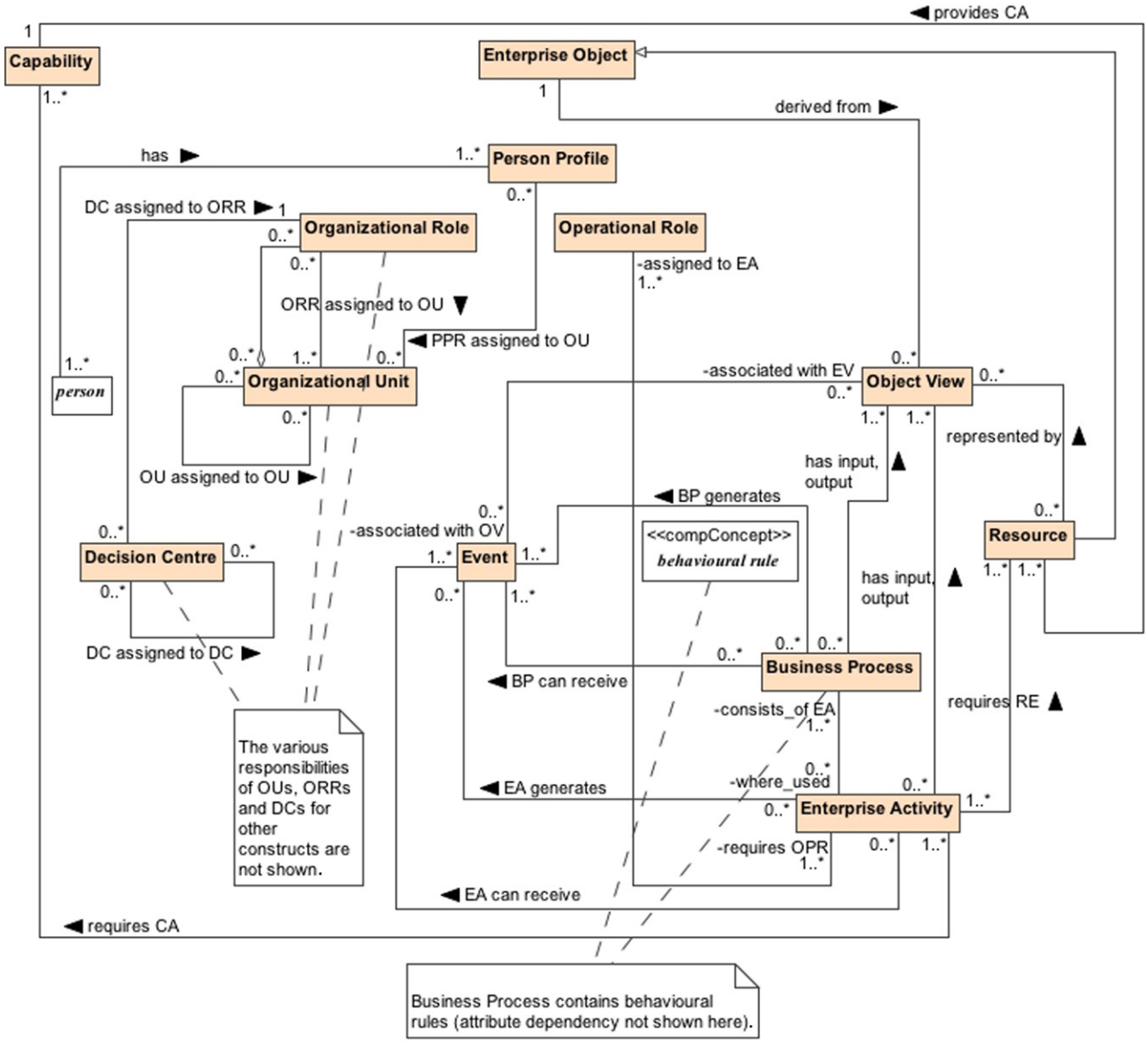

Figure D.12 — Partial metamodel for this International Standard adapted to the ATHENA POP* metamodel

D.3.4 Comparison of definitions

Tables D.3 to D.6 list corresponding terms used in ATHENA POP* and this International Standard, organized by the four POP* dimensions considered. Terms that are relevant to more than one dimension are listed only once.

Table D.3 — List of constructs from this International Standard and corresponding terms from POP* general concept dimension relative to this International Standard

ATHENA POP*	Constructs from this International Standard	Comments
Capability: Capability is a kind of Enterprise Object and is used to designate the quality of being able to do something, like competence, skill and attitude.	Capability: construct that represents the collection of capability characteristics (expressed as capability attributes) of either a Resource (its provided Capability) or an Enterprise Activity (its required Capability).	In this International Standard, Capability refers to a non-human resource. Human skills and competences are modelled by Person Profile, Organizational Role and Operational Role. POP* uses Capability to represent the qualities of any object in the enterprise
Enterprise Object: Enterprise Object may represent anything or anybody that performs work, makes something happen or is just part of some activity in the enterprise. Persons, roles, tangibles (like physical things) and intangibles (like information and knowledge) are included in this concept. An enterprise object takes part in the enterprise through playing roles (instance of Role, Organization Role or Process Role). Enterprise objects may have certain capabilities, and may have a location.	Enterprise Object: construct that represents a piece of information in the enterprise domain that describes a generalized or a real or an abstract entity, which can be conceptualized as being a whole.	This International Standard uses Enterprise Object as a concept to represent information only. Product, Order and Resource are specializations of EOs, whereas POP* uses EO to represent any object in the enterprise.
Location: Location is used to designate a geographic location, a place.	Location: not a construct.	In this International Standard, location can be identified as an attribute of the resource construct (equipment, mfg line, plant, etc.). In POP*, location is a construct in order to be reused for several objects.
Role: The Role concept designates the part played by some object. In most cases, a role is either specified within the context of a process (Process Role) or within the context of an organization unit (Organization Role), but may also be defined in other contexts. Roles may be considered as a placeholder for objects, and are introduced to facilitate representation of important and principle features of a process or an organization, without having to be specific about the concrete objects (persons, departments, etc.) involved.	Organizational Role: construct that represents a set of organizational capabilities, skills and responsibilities that are required by an Organizational Unit and that are to be provided by a person. Operational Role: construct that represents a set of operational capabilities, skills and responsibilities that are required by an Enterprise Activity and that are to be provided by a person.	In this International Standard, the roles to be played by a person are modelled by the Organizational Role and Operational Role; Person Profile is also used to capture the capabilities that a person has. In contrast, the role to be played by a (non-human) resource is modelled by the Resource and the Capability constructs.

Table D.4 — List of constructs from this International Standard and corresponding terms from POP* process dimension relative to this International Standard

ATHENA POP*	Constructs from this International Standard	Comments
Control: Control is a subtype of Process Role. A control is owned by one process.	Control: not used.	Control is somewhat equivalent to a set of Behavioural Rules in this International Standard, which is also owned by a particular process.
Event: representing something that happens at an instant in time. Event should be a specialization of State, with the added requirement that it occurs at a point in time, rather than having a duration.	Event: construct that represents a solicited or unsolicited fact indicating a state change in the enterprise or its environment.	In this International Standard, an Event is a construct. It is a trigger that initiates or is generated by a Business Process or Enterprise Activity.
Flow: A flow is a relationship between two decision points: process roles (input, output, control, or resource) or gateways in a process. Moreover, since decision points are parts of some processes, a flow implicitly expresses a relationship between the processes containing the connected decision points. In the default case, the flow is to be understood as a flow of control. A control flow is used in two different ways: • to indicate temporal ordering between processes, i.e. the sequence in which the processes should be performed, • to indicate the triggering/start-up of the receiving process.	Flow: not used.	In this International Standard, control flow is modelled explicitly, not by enterprise modelling constructs, but by the behavioural rule set, which could be used for the graphical representation of the control flow. Material flow could be derived from the relationships between inputs and outputs of Business Processes and Enterprise Activities.
Input: Input is a subtype of Process Role. An input is owned by a process, and may, as a subtype of Role, represent what is brought into the process. As a subtype of Decision Point, an input represents a condition for starting its “parent” process.	Input: not a construct.	Modelled in this International Standard by input Object View and used in Business Processes and Enterprise Activities.
Output: Output is a subtype of Process Role. An output is owned by a process, and may, as a subtype of Role, represent what is brought out from the Process (e.g. the result). As a subtype of Decision Point, an output represents a condition for continuing after its “parent” process.	Output: not a construct.	Modelled in this International Standard by output Object View and used in Business Processes and Enterprise Activities.

ATHENA POP*	Constructs from this International Standard	Comments
Process: Process is used to represent processes, tasks and activities of any kind. A process can have and be part of a process structure (e.g. work breakdown) through its part of relationship. Moreover, processes may be combined into sequences by means of flows and decision points.	Business Process: construct that represents a partially ordered set of Business Processes or Enterprise Activities, or both, that can be executed to realize one or more given objectives of an enterprise or a part of an enterprise to achieve some desired end-result.	In this International Standard, a Business Process represents processes only; that means it always has Enterprise Activities at its lowest level of decomposition
Process Role: Process Role is both a subtype of Role and a subtype of Decision Point, a fact that indicates the dualistic nature of this concept. As roles, any process role may have enterprise objects attached to it via the plays role relationship. As decision point it represents a condition for continuation of the process in which the process role is included. Process Role is specialized into one of four subtypes: Input, Output, Control, or Resource.	Process Role: not used.	
Resource: Resource is a subtype of Process Role. A resource is owned by a process and, in most cases, resources are used to indicate roles not decision points (although the latter is also possible). As a subtype of Role, Resource is used as a placeholder for various objects used or produced in the process. Examples are Project secretary (representing the actual person having this role), Final report (representing the actual document), Customer (representing the actual person or organization who is waiting for the outcome of the process).	Resource: construct that is a specialization of Enterprise Object construct, which represents the provided capabilities required to execute an Enterprise Activity. Functional Entity: construct that is a specialization of Resource construct, which represents an aggregation of Resources able to perform, completely on its own, a (class of) functional operation(s) required by an Enterprise Activity, or part thereof, and to communicate with the related control.	In this International Standard, Resource is restricted to any non-human means providing the Capabilities that are needed to carry out EAs. The needs have been identified in the required Capabilities. The Functional Entity construct is a Resource capable of undertaking an action.

Table D.5 — List of constructs from this International Standard and corresponding terms from POP* organization dimension relative to this International Standard

ATHENA POP*	Constructs from this International Standard	Comments
Organization Role: Organization Role is a subtype of Role that is defined in the context of an organization. Examples are General manager, Front desk clerk, Department manager, President, etc. In theory, Organization Roles may be played by any type of object, although the typical examples are organization roles played by persons or organization units.	Organizational Role: construct that represents a set of organizational capabilities, skills and responsibilities that are required by an Organizational Unit and that are to be provided by a person.	The difference is that this International Standard defines the capabilities of the person and POP* identifies the role of the person. The capability is covered in the POP* core concept for all enterprise objects.
Organization Unit: Organization Unit is a kind of Enterprise Object used to represent organizations and parts of organizations of any kind. Both formal organization unit and informal, ad hoc groups or teams may be represented by organization units.	Organization Unit: construct that represents an entity of the organizational structure of an enterprise that is described by attributes representing properties of the organization and references to lower level organizational entities.	This International Standard also allows representation of informal organizational entities like teams and groups.
Person: Person designates an individual human being.	Person: not a construct.	This International Standard uses Person Profiles to represent capabilities owned by an individual person.

Table D.6 — List of constructs from this International Standard and corresponding terms from POP* decision dimension relative to this International Standard

ATHENA POP*	Constructs from this International Standard	Comments
Decision Centre: Decision Centre is an abstract concept defined as the set of decisions made at one level and belonging to one decision function. Decision centres are “places” where decisions are taken according to objectives and under constraints. The set of objectives, constraints and variables that limit the freedom for the decision making are collectively referred to as a decision frame. A decision centre corresponds to one or more organization units (instances of Organization Unit). On the other hand, an organization unit may cover more than one decision centre.	Decision Centre: construct that represents a set of decision-making activities that are characterized by having the same time horizon and planning period and belonging to the same kind of functional category.	this International Standard also requires the link to an Organization Unit. Therefore both definitions are equivalent.
Decision Frame: A decision frame is defined as a decisional link between two decision centres. A structural decision frame describes for a decision centre the frame within which it can make a decision. To avoid conflicts, a decision centre is under the influence of only one structural decision frame.	Decision Frame: is not a construct, although three of the Decision Centre attributes are collectively called a *decision frame.*	

Annex E
(informative)

Example usages of constructs and complementary concepts

E.1 Purpose of this annex

Clause E.2 describes where each modelling language construct is typically used in different model phases. Clause E.3 describes where each complementary concept is used in the various construct templates of Clause 6, and Clause E.4 describes how templates can be used in model development. The purpose of this annex is to augment the descriptions of each construct in Clause 6 and the UML models of Annex C by providing additional illustrative detail.

E.2 Modelling language constructs identified for the different model phases

The following lists typical usages for those constructs whose identification is phase-dependent:

— **domain identification:**

  — Domain, Business Process (to identify the main functionality needed to transform Domain inputs into Domain outputs), Enterprise Object and Object View (to identify those Domain Inputs and Outputs), Events, Organizational Role and possibly Organizational Unit (both the latter to identify design authority).

— **concept definition:**

  — Domain, (Business Process, Enterprise Object, Event, Object View, Organizational Unit)[46], Decision Centre.

— **requirement definition:**

  — Domain[47], Business Process, Enterprise Activity, Capability (required), Enterprise Object and Object View, Organizational Unit, required Organizational Role, required Operational Role, Decision Centre.

— **design specification:**

  — Domain[48], Business Process, Enterprise Activity, Event, Enterprise Object Product, Order and Object View Resource, Functional Entity, Capability (provided), Organizational Unit, provided Organizational Role, provided Operational Role, Person Profile, Decision Centre.

— **implementation description:**

  — All of the above to capture any deviation of the implemented system components from the design specifications.

46) As for the domain identification, only its identification is required at this concept identification phase.

47) No additional information required at this requirement definition phase.

48) No additional information required at this design specification phase.

— **domain operation:**

  — Particular model for model-based decision support and operational monitoring and control.

— **decommission definition:**

  — All of the above to identify decisions on future use of the operational system components.

E.3 Complementary concepts and their usages

Each complementary concept is listed in 5.7. The following lists in which construct templates they are used and provides additional information where appropriate.

Behavioural rules are used to specify the logical sequencing of a Business Process in terms of its constituent activities, initiating events and activity states. They are introduced in 6.3.5 and defined in detail in Annex A. In the templates of Clause 6, *behavioural rules* appear within Business Process (behavioural rule set). They do not have an associated identifier.

Constraints are textual fragments or first-order predicates stating a restriction or limitation or condition placed upon most of the constructs. They are used by the following templates: Domain, Business Process, Enterprise Activity, Enterprise Object, Object View, Product, Order, Resource, Capability, Functional Entity and Decision Centre. They are also used within *declarative rules*, defined by *integrity rules* when applied to Enterprise Objects, and defined by relationship-related rules when applied to relationships and transactions (see 5.5). *Constraints* may have an associated identifier.

Declarative rules are used in Business Process and contain a set of *objectives* and *constraints*. Each *declarative rule* has an associated name or identifier or both.

Functional operations are used in Enterprise Activity (Consists_of) and Functional Entity (Operation Set). They are described in 6.4.5. Each *functional operation* has an associated name, a reference to a Functional Entity and a list of arguments including a status code.

Integrity rules are used in Enterprise Object, Object View, Product, Order, Resource, Capability and Functional Entity during the requirements definition phase to make statements of the “existence of information and information conformity to real-world reality” (from the definition). Such statements are interpreted in later enterprise model phases as *constraints* on information. An *integrity rule* has an associated name or identifier or both.

Objectives are textual fragments stating strategic and operational *objectives* (and the relationships between them) to be achieved in a Domain, a Business Process, an Enterprise Activity or a Decision Centre. They are used in Domain, Business Process, Enterprise Activity and Decision Centre (part of the decision frame). They are also used in d*eclarative rules*. *Objectives* may have an associated identifier.

Performance indicators are textual fragments defining measures or metrics to be used in assessing whether o*bjectives* have been achieved. They are used in Domain, Business Process and Enterprise Activity.

E.4 Using templates in model development

E.4.1 An illustrative example

This example of order processing in a manufacturing enterprise has been taken from ISO 19439:2006, B.3. It has been extended to show the use of templates in enterprise modelling.

An overview of the example is presented in Figure E.1 showing the enterprise entities in the form of an entity-relation diagram, which is organized according to the view concept for the four views defined in the standard: function, information, resource and organization view (see Figures E.2 to E.5 for more details on individual views)

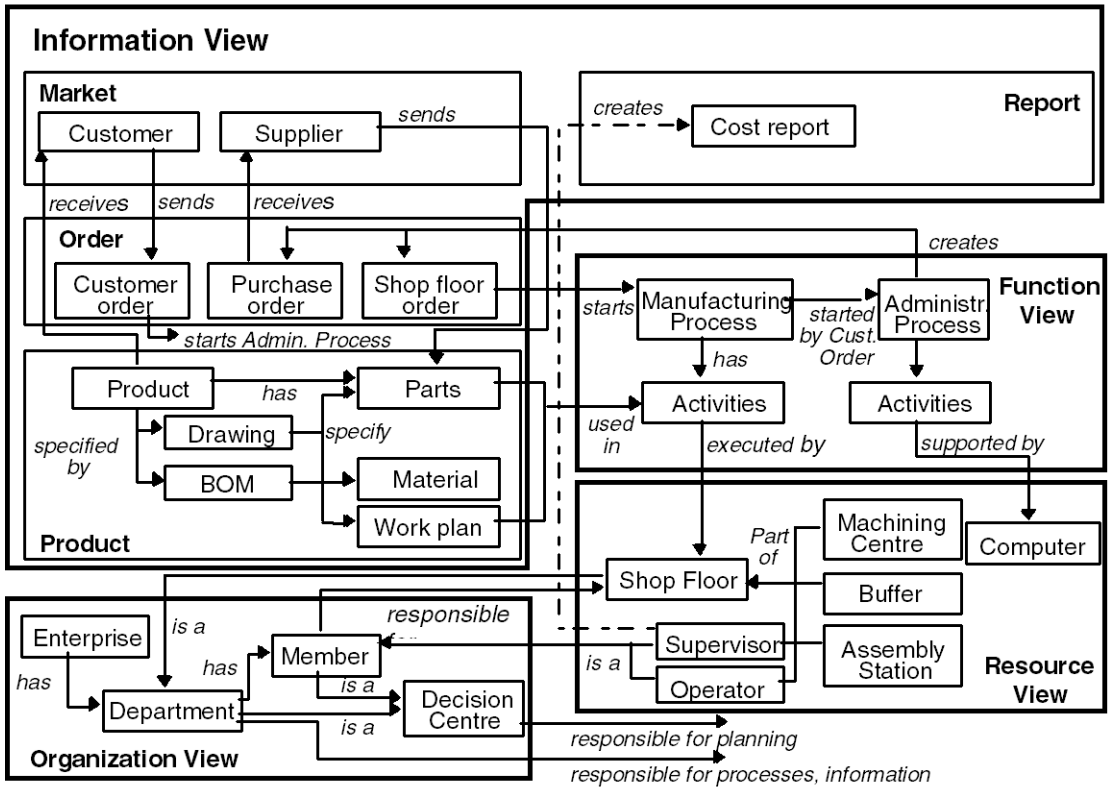

Figure E.1 — Model views for order processing

The function view represents the operational processes (Production and Administration) and the associated activities. Activity inputs and outputs are represented and structured in the information view. The information objects used in the order processing are organized in a set of objects (Market, Order, Product and Report), which, in general, are composed of other objects. The different enterprise objects and their constituents have relations to other objects in the same or other views, which are identified by the directed links between the different objects.

In the resource view, the required resources are identified and organized into composed structures (Shop Floor) that may follow the organization structure of the enterprise (Shop Floor is a department). Person Profiles describe the operational and organizational skills of the people involved (Supervisor and Operator) and identify their relations to the organization view.

The organization view represents the relations between the different organizational entities in the enterprise and identifies their responsibilities. These responsibilities can be on different levels of the organization and are usually carried out by members of the organization entity authorized for the decision making. The GRAI concept[12] of decision centres addresses this decision making aspect. Identifying the time horizons of the different decision centres then qualifies the planning aspect of decision making. The contents of the different views and the corresponding templates are presented in the following sections and summarised in two tables in the summary section.

E.4.2 Function view

E.4.2.1 Development of the Function View

Figure E.2 shows further details of the function view introduced in the example above. Two of the three identified domains (Customer and Supplier) provide inputs (Customer/Order, Supplier/Parts) to, and receive outputs (Customer/Product & Invoice, Supplier/Order & Payments) from, the Domain Own Enterprise. The latter consists of the two processes (Administration and Manufacturing) identified in the function view of Figure E.2. Processing of the domain is started by the Event EV-1 Arrival of Customer Order, which has

associated the order itself in the form of an Object View OV-1, and which starts BP-1/Administration (see Event Template below).

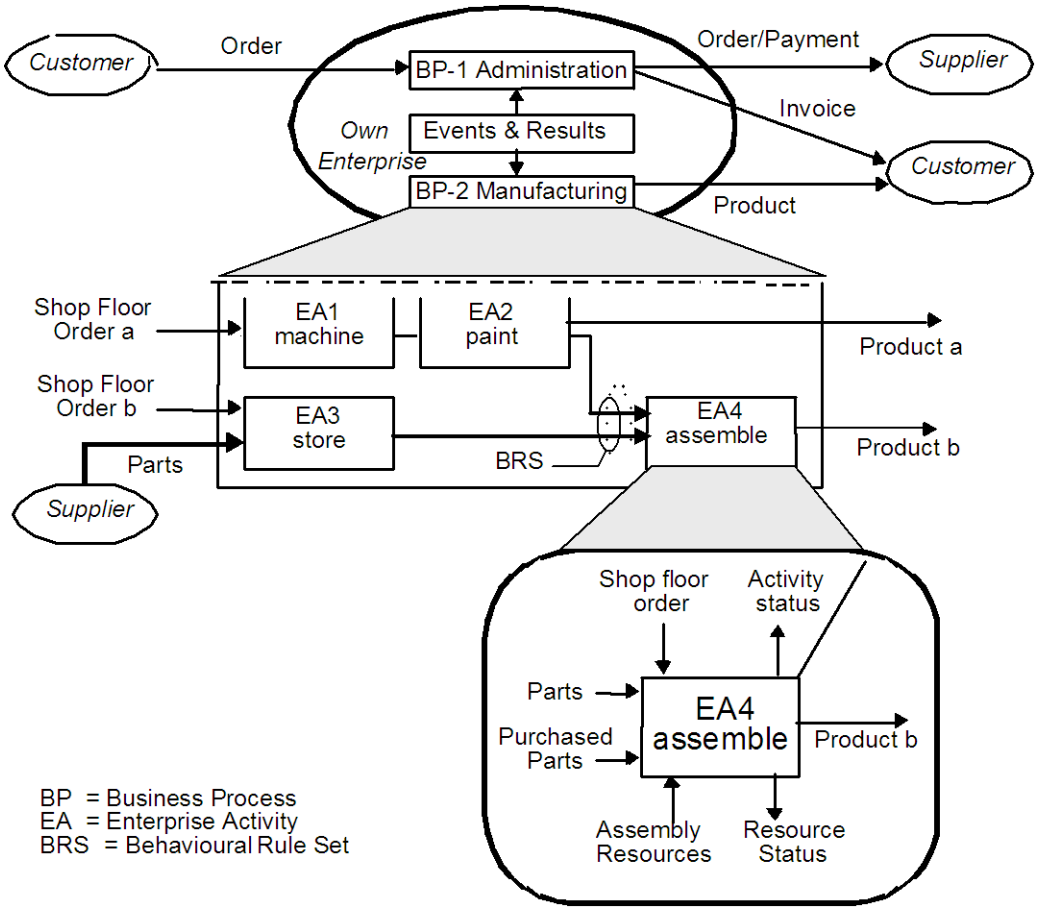

Figure E.2 — Function view for order processing

Decomposition of BP-2 Manufacturing leads to the network of four enterprise activities shown in the centre part of Figure E.2. The lower part of Figure E.2 shows the links to the objects in the information and resource views that are identified as inputs and outputs of the enterprise activity EA4. Not shown are links to the organization view made via responsibilities or identification of organizational aspects (Shop Floor, treated as a combination of Department and Decision Centre). In the following examples of the templates, the four constructs, Domain, Business Process, Enterprise Activity and Event, which are relevant for the function view, are presented.

E.4.2.2 Example construct template for Domain

Header	
Construct label	DM
Identifier	DM-1
Name	Manufacturing Order Processing (Own Enterprise Domain)
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	identifies all domain inputs/outputs and internal functionalities necessary for order processing starting with Receive Customer Order and Ending with Receive Customer Payment.
Business Process Description	BP-1 / Administration: handles all needed orders, invoices and payments BP-2 / Manufacturing: handles all needed material processing and transports

The following five descriptives are described at a level of detail appropriate to the enterprise model phase		
Objectives[49)]	strategic: achieve customer satisfaction, increase market share, …	
	operational: satisfy customer demand, …	
Constraints	resource availability, regulations on working time and environment	
A2 Descriptives relevant for different enterprise model phases		
A2.1 applicable at concept definition and later phases		
Domain Characterization	Mission: special equipment – manufacture to order	
	Vision: increase market share through collaboration	
	Values: honest, fair and trustworthy behaviour to customers, personnel and community	
Domain Operation	Strategies: to increase product range through collaboration with x,y,z	
	Policies: enhance own core competencies through hiring	
	Operational concepts: reconfigure mfg. lines for increased flexibility	
	Business plans: introduce new product capabilities	
Decisional Authority		
Decision Function	DC-1 / Plan Production	
Decision Level	TBD[50)]	
A2.2 applicable at requirement definition and later phases		
Performance Indicators	on time delivery; number and content of customer complaints	
B1 Relationships relevant for all enterprise model phases		
Inputs/Outputs		
Object View Inputs	DM-X / Customer :	OV-1 / Customer Order;
	DM-X / Customer :	OV-2 / Customer Payment;
	DM-Y / Supplier :	OV-3 / Purchased Parts;
	DM-Y / Supplier :	OV-4 / Invoice;
Event Inputs	DM-X / Customer :	EV-1 / Arrival Customer Order;
	DM-X / Customer :	EV-2 / Arrival Customer Payment;
	DM-Y / Supplier :	EV-3 / Arrival Purchased Parts;
	DM-Y / Supplier :	EV-4 / Arrival Invoice;
Object View Outputs	OV-5 / Product a :	DM-X / Customer;
	OV-6 / Invoice :	DM-X / Customer;
	OV-7 / Purchase Order :	DM-Y / Supplier;
	OV-8 / Payment :	DM-Y / Supplier;
	OV-11 / Product b :	DM-X / Customer;
Event Outputs	EV-5 / Shipment Product :	DM-X / Customer;
	EV-6 / Sending Invoice :	DM-X / Customer;
	EV-7 / Sending Order :	DM-Y / Supplier;
	EV-8 / Sending Payment :	DM-Y / Supplier;
B2.2 applicable at requirements definition and later phases		
Business Processes	BP-1 / Administration; BP-2 / Manufacturing	
B2 Relationships relevant for different enterprise model phases		
B2.3 applicable at design specification and later phases		
Operational Relationships		
Operation Responsibility	OU-1 / Enterprise Operation	
Operation Authority	OU-1 / Enterprise Operation	

49) Objectives occur in Business Processes as well; consistency has to be assured.

50) Here and in the remainder of E.4.2, TBD denotes that additional information is required that is not provided in this illustrative example.

E.4.2.3 Example construct template for Business Process

Header		
Construct label	BP	
Identifier	BP-2	
Name	Manufacturing	
Design Authority	OU-2 / Production Planning	
Body		
A1 Descriptives relevant for all enterprise model phases		
Description	defines all manufacturing activities needed for material handling, processing and transportation	
Objectives	strategic: increase profitability operational: increase operational flexibility of mfg. line through workforce training and new technology.	
A.2 Descriptives relevant for different enterprise model phases		
A2.2 applicable at requirements definition and later phases		
Constraints	order volume, resource availability, regulations on working time and environment	
Performance Indicators	TBD	
Declarative Rules	TBD	
Behavioural Rule Set	WHEN (START AND EV-9) DO EA1 ; WHEN (ES(EA-1) [51] = OK) DO EA-2 ; WHEN (ES(EA-1) <> OK) DO EXCEPTION e-1 ; WHEN (ES(EA2) = m) DO FINISH AND EV-11 ; WHEN (ES(EA2) = n) AND (ES (EA-3) = OK) DO EA-4 ; WHEN (ES(EA2) <> n) DO EXCEPTION e-2 ; WHEN (ES(EA-3) <> OK) DO EXCEPTION e-2 ; WHEN (START AND EV-3) AND (EV-10) DO EA3 ; WHEN (ES(EA-3) = OK) AND (ES (EA2) = n) DO EA-4 ; WHEN (ES(EA-3) = OK) AND (ES (EA2) <> n) DO EXCEPTION e-3 ; WHEN (ES(EA4) = OK) DO FINISH AND EV-5 ;	
A.2.3 applicable at design specification and later phases		
Priority	TBD	
B1 Relationships relevant for all enterprise model phases		
Where_used	DM-1 / Manufacturing Order Processing (Own Enterprise Domain)	
B2 Relationships relevant for different enterprise model phases		
B2.2 applicable at requirements definition and later phases		
Part_of	NIL	
Consists_of	EA-1 / machine; EA-2 / paint; EA-3 / store; EA-4 / assemble	
Inputs/Outputs	*Origin/Object View*	*Object View/Destination*
Object View Inputs	DM-Y / Supplier :	OV-3 / Purchased Parts;
	BP-1 / Administration :	OV-9 / Shop Floor Order a;
	BP-1 / Administration :	OV-10 / Shop Floor Order b;
Event Inputs	DM-Y / Supplier :	EV-3 / Arrival Purchased Parts;
	BP-1 / Administration :	EV-9 / Shop Floor Order a;
	BP-1 / Administration :	EV-10 / Shop Floor Order b;
Object View Outputs	OV-5 / Product a :	DM-X / Customer;
	OV-11 / Product b :	DM-X / Customer;
Event Outputs	EV-5 / Shipment Product :	DM-X / Customer, BP-1 / Administration
	EV-11 / Completion Product a :	BP-1 / Administration
B2.3 applicable at design specification and later phase		
Operational Relationships		
Operation Responsibility	OU-3 / Shop Floor Department	
Operation Authority	OU-3 / Shop Floor Department	

51) ES(…) is an abbreviation for Ending Status (…).

E.4.2.4 Example construct template for Enterprise Activity

Header		
Construct label	EA	
Identifier	EA-4	
Name	assemble Product	
Design Authority	OU-2 / Production Planning	
Body		
A1 Descriptives relevant for all enterprise model phases		
	Not applicable	
A2 Descriptives relevant for different enterprise model phases		
A2.2 applicable at requirements definition and later phases		
Description	identifies all inputs and outputs needed for the task of product assembly starting with part handling and ending with product completion	
Activity Behaviour	TBD	
Objectives	strategic: improve assembly operation (relates to: 'increase profitability'); operational: re-engineer assembly operation, motivate operator (relates to: 'increase operational flexibility of mfg. operation through workforce training and new technology')	
Constraints	resource availability, regulations on working time and environment	
Performance Indicators	turn around time; work in process, product cost	
A2.3 applicable at design specification and later phases		
Consists_of	FO-1 / move part (part/product); FO-2 / position part; FO-3 / fix part; FO-4 / test product	
Inputs/Outputs	*Origin/Object View*	*Object View/ Destination*
Ending Statuses	ES(EA-4) <OK>; ES(EA-4) <EXCEPTION4>	
Duration	TBD	
B1 Relationships relevant for all enterprise model phases		
	Not applicable	
B2 Relationships relevant for different enterprise model phases		
B2.2 applicable at requirements definition and later phases		
Where_used	BP-2 / Manufacturing	
Inputs/Outputs	*Origin/Object View*	*Object View/Destination*
Function Inputs	PR-1.1 / Part :	OV-12 / painted Part;
	PR-1.1 / Part :	OV-13 / purchased Part;
Control Inputs	BP-1 / Administration :	OV-10 / Shop Floor Order b;
Required Operational Role	OPR 2 / Operator;	PPR-2 / Operator
Required Capabilities	CA-1 / Mechanical Product Assembly :	EA 4 / assemble Product
Function Outputs	OV-5 / Product b :	DM-X / Customer;
Input Events	NIL	
Output Events	EV-5 / Shipment Product b :	DM-X / Customer, BP-1 / Administration;
	EV-12 / Exception4 :	Exception Handling (TBD)
B2.3 applicable at design specification and later phases		
Inputs/Outputs	*Origin/Object View*	*Object View/Destination*
Operational Role Outputs	OV-13 / Assembly Information (OPR-3)	EO-6 Report
Control Outputs	OV-14 / Status Information (EA-4) :	EO-4 / Report; and EO-5 / Activity Status;
Resource Inputs	RE-1.4 / Assembly Resources; CA-2 / Mechanical Product Ass. Station;	
	PPR-2 / Operator	OPR-2 / Operator A. Smith;
Resource Outputs	OV-15 / Resource Information (EA-4) :	EO-2 Report and EO-4 Resource Status;
Operational Relationships		
Operation Responsibility	OU-3 / Shop Floor Department	
Operation Authority	OU-3 / Shop Floor Department	

E.4.2.5 Example construct template for Event

Header	
Construct label	EV
Identifier	EV-1
Name	Arrival Customer Order
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Initiates BP-1 / Administration leading to generation of Shop Floor Orders and Purchase Order
A2 Descriptives relevant for different enterprise model phases	
A2.3 applicable at design specification and later phases	
Timestamp	TBD
Priority	TBD
B1 Relationships relevant for all enterprise model phases	
Object Views	OV-1 / Customer Order
Generated_by	DM-X/ Customer
Initiates	BP-1 / Administration
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-1 / Enterprise Operation
Operation Authority	OU-1 / Enterprise Operation

E.4.3 Information view

E.4.3.1 Development of the Information View

Figure E.3 identifies the relations between the information view and the function view for the assemble activity showing only relations for function inputs (Parts and Purchased Parts) and outputs (Product).

All the information used and produced by the Enterprise Activity as Object Views is or becomes part of the related Enterprise Objects. Parts and Product are represented in the information view as objects. Only projections of these objects called Object Views (CIMOSA[11]), which select a limited set of information of the object, are used in the function view. As indicated already in figure E.1, the Enterprise Object *Part* is part of the Enterprise Object *Product*. For the remainder of the information view, only the composed objects and their relations are shown in Figure E.3.

The concept of Object View requires multiple use of information as indicated in the two Object Views derived from the Enterprise Object Part (both Object Views use attributes 1 and 2 together with additional attributes). However, all of these Object Views are of a temporary nature, which means they only exist together with an Enterprise Activity instance and this information redundancy is of no consequence for the consistency of the overall information model represented in the information view.

The information view can be used in two different ways as indicated by the links with the two arrows.

a) The necessary Object Views can be identified according to the needs of the operational activities and the Enterprise Objects are assembled from the multitude of object projections identified in the course of modelling.

b) The Object Views are selected from the predefined Enterprise Objects.

Option b) is the preferred way of providing consistency of the enterprise model and the enterprise knowledge base.

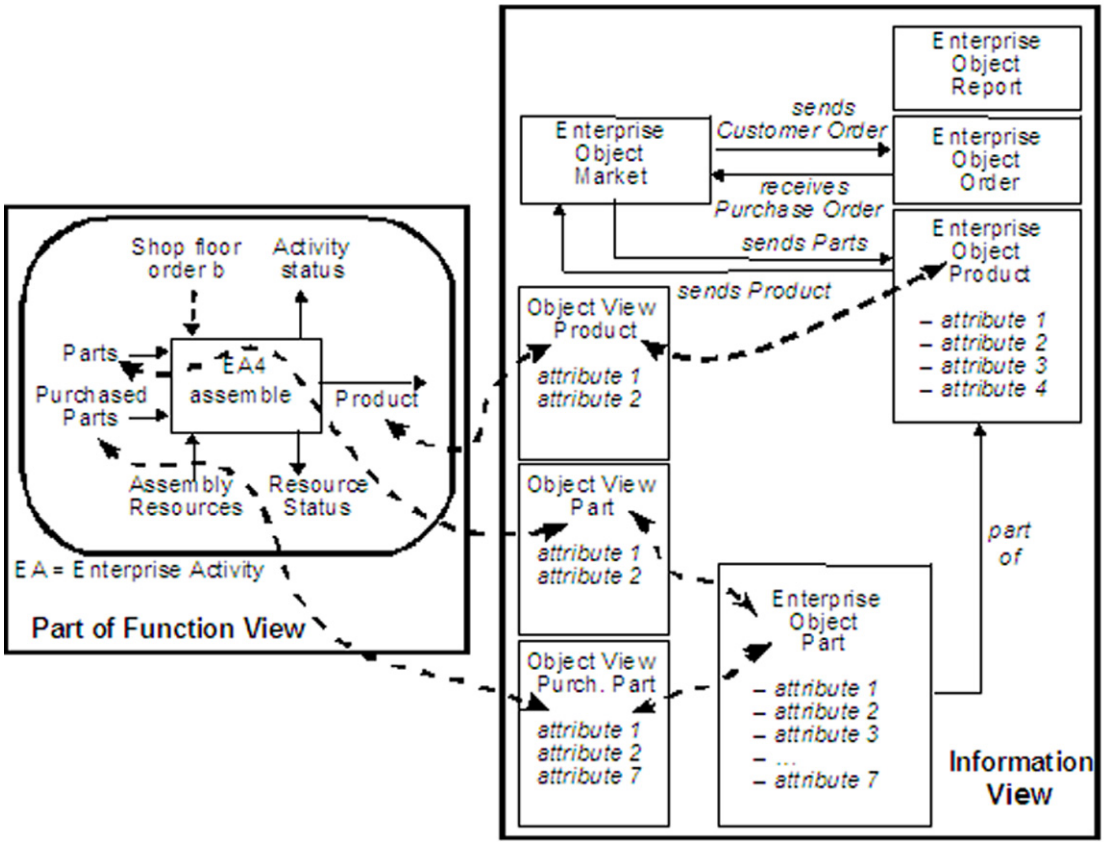

Figure E.3 — Information view for order processing

E.4.3.2 Example construct template for Enterprise Object

Header	
Construct label	EO
Identifier	EO-1.1
Name	Customer
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Contains all information about the customer
Nature of Object	INFORMATION
Properties	Customer Information
Constraints	NIL
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Enterprise Object relationships in the form:	
Specialization_of	NIL
Part_of:	NIL
Consists_of:	
Related_to	EO-1.1.1 / Customer Payment [1..n]; EO-1.1.2 / Customer Invoice [1..n]
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-4 / ICT Department
Operation Authority	OU-1 / Enterprise Operation

E.4.3.3 Example construct template for Object View

Header	
Construct label	OV
Identifier	OV-13
Name	purchased Part
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Contains all part information needed for the assembly operation
Properties	part id = ABC123; part geometry = TBD; part material;=steel = ST 123
Constraints	NIL
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
Enterprise Object	PR-1.1 / Part [1..n]
Events	NIL
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-4 / ICT Department
Operation Authority	OU-3 / Enterprise Operation

E.4.3.4 Example construct template for Product

Header	
Construct label	PR
Identifier	PR-1
Name	Product
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Contains all information on the product
Nature of Object	PHYSICAL
Properties	Product Information
Constraints	NIL.
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Product relationships in the form:	
Specialization_of	NIL
Part_of:	NIL
Consists_of:	PR-1.1 / Part [1..n]; PR-1.2 / Drawing [1..n]; PR-1.3 / BOM [1..n]
Related_to	EO-1.1 / Customer [1..n]; OR-1.1 / Customer Order [1..n]
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-4 / ICT Department
Operation Authority	OU-3 / Enterprise Operation

E.4.3.5 Example construct template for Order

Header	
Construct label	OR
Identifier	OR-1
Name	Order
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Contains all information on the orders
Nature of Object	INFORMATION
Properties	Order Information
Constraints	NIL
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
Class Relationships – list of Order relationships in the form:	
Specialization_of	NIL
Part_of:	NIL
Consists_of:	OR 1.1 / Customer Order [1..n]; OR 1.2 / Purchase Order [1..n]; OR-1.3 / Shop Floor Order [1..n]
Related_to	EO-1.1 / Customer [1..n]; EO-1.2 / Supplier [1..n]; PR-1 / Product [1..n];
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-4 / ICT Department
Operation Authority	OU-3 / Enterprise Operation

E.4.4 Resource view

E.4.4.1 Development of the Resource View

Figure E.4 identifies the relations between the resource view and the function view using only assemble activity resources. Again, the concept of Object Views is employed. As indicated already in Figure E.1, the Resource Object *Assembly Station*, together with other equipment resources, is part of the Resource Object *Shop Floor*. The operator involved in the assembly operation is represented by the Person Profile *Operator*. Not shown is the Functional Entity *Handling*, which consists of the *Operator* identified in Figure E.4 and a robot.

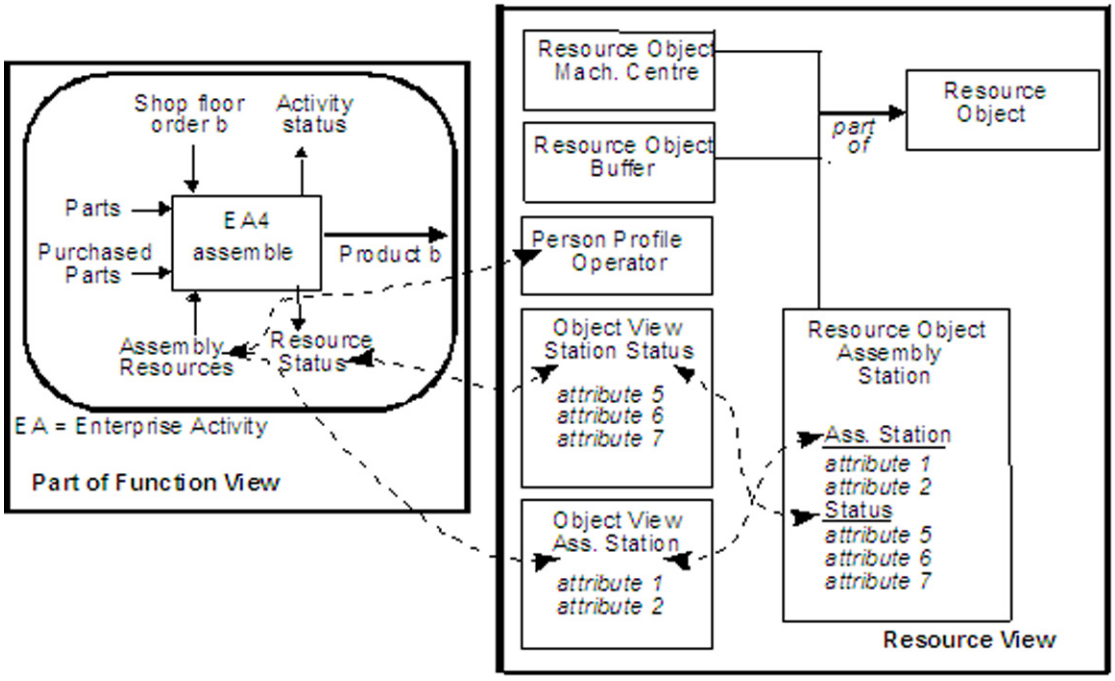

Figure E.4 — Resource view for order processing

Object Views of the Assembly Resources and of the Operational Role of the operator describe the capabilities of the station itself and those of the operator, one each for the operator and the resources of the assembly station, and one each for the operator and for the status of the station after completion of the assemble activity. The latter provides for information such as operation duration, available lifetime of time-critical components and identification of maintenance operations needed.

The resource view can also be used in two different ways as indicated by the various double-headed arrow links.

a) The necessary object projections can be identified according to the needs of the operational activities and the Resource objects are assembled from the multitude of object projections identified in the course of modelling.

b) The object projections are selected from the predefined Enterprise Objects.

Option b) is the preferred way of providing consistency of the enterprise model and the enterprise knowledge base.

E.4.4.2 Example construct template for Resource

Header	
Construct label	RE
Identifier	RE-1.4
Name	Assembly Resources
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
	Not applicable
A2.3 applicable at design specification and later phases	

Description	contains all information on the assembly resources needed to assemble mechanical products
Nature of Object	PHYSICAL
Properties	location Building 123, installed July 2003, book_value €20,000.
Constraints	MTF >1500 hours, MTR ≤ 4 hours
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Provided Operational Roles	OPR-3 / Assembly line operator
Provided Capability	CA-2 / Mechanical Product Assembly Station
Class Relationships – list of Resource relationships in the form:	
Specialization_of	NIL
Part_of:	RE-1 Shop Floor
Consists_of:	RE-1.4.1 / Robot; RE-1.4.2 / Test Station; RE-1.4.3 / Assembly Fixture RE-1.4.4 / Assembly Tools
Related_to	PR-1 / Product [1..n]
Operational Relationships	
Operation Responsibility	OU-3 / Shop Floor Department
Operation Authority	OU-3 / Shop Floor Department

E.4.4.3 Example construct template for Capability

Header	
Construct label	CA
Identifier	CA-2
Name	Mechanical Product Assembly Station
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Description	contains all information needed to describe a station for assembling mechanical products
Function-related Attributes	handling part and product: from pallet into assembly station, assembly station into test station, test station onto product pallet (distance up to 1 m); positioning: part into assembly fixture, product into test station, product on pallet (accuracy ± 0.1 mm); product dimension (up to.2m^3); product weight (up to 20 kg) throughput_assembling ≤20/hour); throughput_testing ≤ 20/hour;
Constraints	availability > 99.9%
Integrity Rules	NIL
A2.3 applicable at design specification and later phases	
Performance-related Attributes	20 products/hour
Operation-related Attributes	2 shift operation
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.2 applicable at requirements definition and later phases	
Capabilities Included	CA-2.2 / Robot;
Where_required	EA-4 / assemble Product
Where_provided	TBD
B2.3 applicable at design specification and later phases	
Operational Relationships	
Operation Responsibility	OU-3 / Shop Floor Department
Operation Authority	OU-3 / Shop Floor Department

E.4.4.4 Example construct template for Functional Entity

Header	
Construct label	FE
Identifier	FE-1
Name	Handling
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.3 applicable at design specification and later phases	
Description	contains all relevant information of the Functional Entity
Nature of Object	PHYSICAL
Properties	ICT capabilities: PC controlled, LAN
Operation Set	Functional Operation (FO)-1 move Part (part pallet to fixture); FO-2 move Product (fixture to test station, test station to product pallet); FO-3 position Part in fixture; FO-4 position Product (in Test Station, on product pallet)
Constraints	MTF/MTR, Movement range ≤1.2m; Positioning accuracy: ± 0.1mm; Pay Load: ≤25 kg; Operating cost: ≈100 €/hour;
Integrity Rules	NIL
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Provided Operational Role	OPR-123 / Handling station operator
Provided Capabilities	CA-2.2 / Robot;
Class Relationships – list of Functional Entity relationships in the form:	
Specialization_of	NIL
Part_of:	RE-1.4 / Assembly Resources
Consists_of:	NIL
Related_to	PR-1 / Product [1..n]
Operational Relationships	
Operation Responsibility	OU-3 / Shop Floor Department
Operation Authority	OU-3 / Shop Floor Department

E.4.5 Organization view

E.4.5.1 Development of the Organization View

Figure E.5 shows the organizational and decisional aspects of the illustrative example of order processing. Starting from the general concept shown in the lower left corner of Figure E.1, the responsibilities of the different departments and its members are indicated for the objects or Object Views in the other views (function, information and resource).

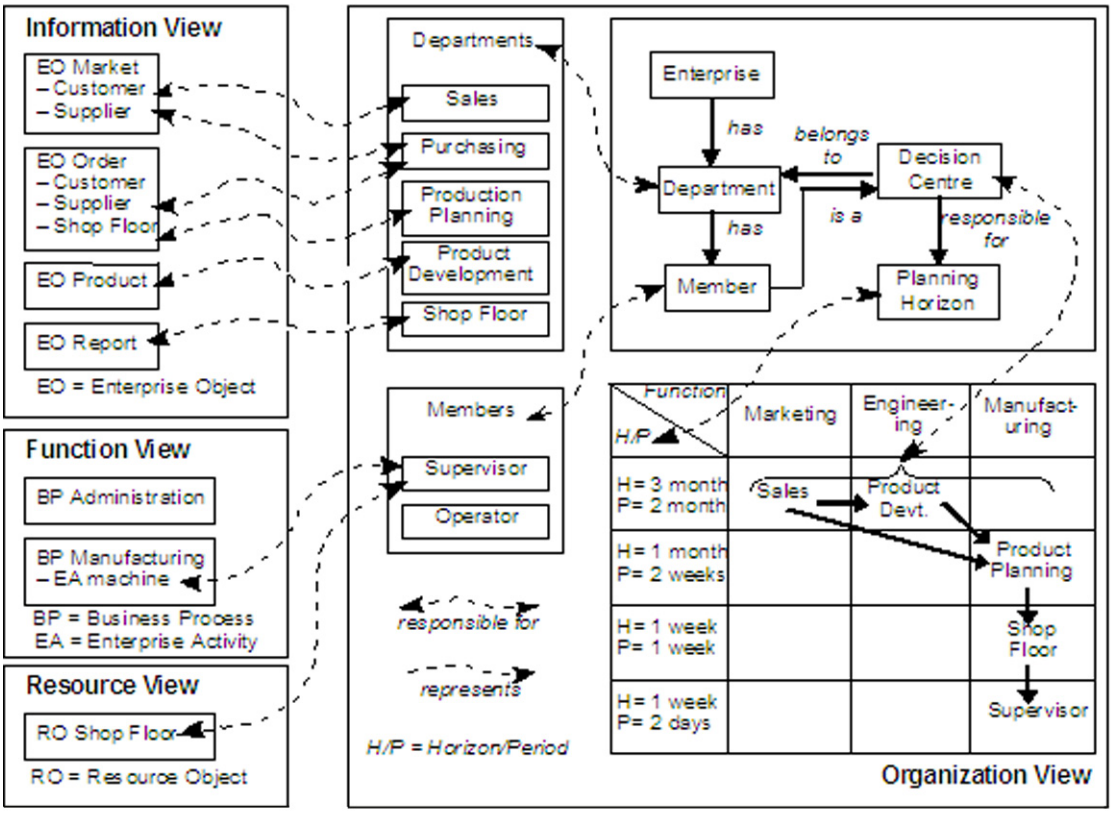

Figure E.5 — Organization view for order processing

In the lower right corner of Figure E.5 the GRAI grid[12] is used to represent the relations between different decision centres (indicated by the arrows). Decision centres are ordered in the grid according to the duration of their planning horizons and the associated period for revision of the decision.

E.4.5.2 Example construct template for Organizational Unit

Header	
Construct label	OU
Identifier	OU-3
Name	Shop Floor Department
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
Description	Responsible for shop floor operation
A2 Descriptives relevant for different enterprise model phases	
A2.3 applicable at design specification and later phases	
Organization Level	first level department
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Authorities and Responsibilities	
Process-related authority/ responsibilities:	BP-2 / Manufacturing
Information-related authority/	NIL

responsibilities:	
Resource-related authority/ responsibilities:	RE-1 / Shop Floor Resources
Assignments	
Assigned Organization Roles/ Units	ORR-1 / Supervisor Shop Floor;
Assigned to Organization Units	OU-1 / Enterprise Operation

E.4.5.3 Example construct template for Decision Centre

Header	
Construct label	DC
Identifier	DC-1
Name	Production Overload Levelling
Design Authority	[OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
Description	Short-term production planning in order to level overload
Decision Frame (defining the decision context to be satisfied by the Decision Centre instance)	
Objectives	minimise external sub-contracting
Variables	extra_hours; sub-contracting_hours
Constraints	50< extra_hours <100; 100< sub-contracting_hours <500
Decision Category	Plan Production
Decision Level	H = 2 weeks, P = 1 week
A2.3 applicable at design specification and later phases	
Organization Level	second level
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Operational Applicability	
Process applicability	BP-2 /Manufacturing
Information applicability	EO-1 / Market; EO-2 / Report; EO-3 / Status; OR-1 / Order; PR-1 / Product;
Resource applicability	RE-1
Assignments	
Assigned Decision Centres	NIL
Assigned to Decision Centre	NIL
Assigned to Organization Role	ORR-1 / Supervisor Shop Floor
Assigned to Organization Unit	OU-3 / Shop Floor Department

E.4.5.4 Example construct template for Person Profile

Header	
Construct label	PPR
Identifier	PPR-1
Name	John Smith
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.3 applicable at design specification and later phases	

Employee Profile relevant for the employment in the enterprise	
Organization-related Skills	Leadership qualities; Experience in Operational planning; Department budgeting;
Operation-related Skills	Experience in mechanical part manufacturing;
Role Job Description	Manager of Shop Floor Department
Provider of Skills	ID1234 / John Smith;
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.3 applicable at design specification and later phases	
Assignments	
Assigned to Organization Unit	OU-3 / Shop Floor Department
Assigned to Organizational Role	ORR-1 7 Supervisor
Assigned to Operational Role	NIL

E.4.5.5 Example construct template for Organizational Role

Header	
Construct label	ORR
Identifier	ORR -1
Name	Supervisor Shop Floor
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.3 applicable at requirement definition and later phases	
Organizational Skills	Leadership; Operational planning; Department budgeting;
Responsibilities	BP-2 / Manufacturing; Operation of Shop Floor Department
Authorities	Internal Organization of Shop Floor Department; Distribution of workload
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.2 applicable at requirements definition and later phases	
Specialization_of	NIL
B2.3 applicable at design specification and later phases	
Assignments	
Assigned to Organization Unit	OU-3 / Shop Floor Department
Assigned Decision Centre	NIL

E.4.5.6 Example construct template for Operational Role

Header	
Construct label	OPR
Identifier	OPR-2
Name	Operator
Design Authority	OU-2 / Production Planning
Body	
A1 Descriptives relevant for all enterprise model phases	
	Not applicable
A2 Descriptives relevant for different enterprise model phases	
A2.2 applicable at requirements definition and later phases	
Role Job Description	Operate and maintain assembly work station
Operational Skills	Part and Product handling; PC and Robot operational knowledge
A2.3 applicable at design specification and later phases	

Role Job Description	Operator of Assembly Work Station
Operational Profile	Mechanical product assembly, Assembly station set-up and tear-down, robot operation, PC operation, Visual inspection and test operation;
B1 Relationships relevant for all enterprise model phases	
	Not applicable
B2 Relationships relevant for different enterprise model phases	
B2.2 applicable at requirements definition and later phases	
Specialization_of	NIL
B2.3 applicable at design specification and later phases	
Assignments	
Assigned to Organization Unit	OU-3 / Shop Floor Department
Assigned to Enterprise Activity	EA-4 / assemble

E.5 Template Use — Example Contents Summary

The use of the modelling language constructs during model progression is shown in Table E.1, using the Order Processing example of Figure E.1. For each model phase, the most important steps of the modelling process and the related modelling language constructs are identified. This identification starts with the Domain construct in the Domain Identification, which enables the definition of domain boundaries (Domain inputs and outputs with their origins and destinations) and its internal functionality needed to transform inputs into outputs, respectively. In the following phases, additional information captures domain missions, etc. (Concept Definition phase) and adds details to inputs/outputs and functionality. Business Process, Enterprise Object, Object View, Event and Organizational Unit constructs complement the Domain construct, providing model relevant identification of domain functionalities, its inputs and outputs and last but not least model design authority.

In the Requirements Definition Phase, the two constructs Business Process and Object View, together with the Enterprise Activity, Capability and Operational Role construct, allow capturing the operational requirements for the domain, using and extending the information captured at previous phases. Specific information on the start conditions of a Business Process can be described by means of the Event construct. The Enterprise Object construct allows the information items to be described from which the Object Views are derived or into which they are composed. The organizational aspects of the domain can be captured by a network or hierarchy of Organizational Units and Decision Centres using the Organizational Role construct to describe the human related requirements.

The Design Specification phase describes solutions for the domain requirements using all the constructs identified above and capturing the detailing information needed for the operation design. From the required capabilities defined in the Enterprise Activity construct, the specifications of resource capabilities will be derived and captured in the Resource construct. To enable a consistent resource assignment, the functionality of the Enterprise Activity is partitioned into Functional Operations, in such a way that they are executable by an autonomous resource which is described by the Functional Entity construct. The Person Profile construct is then introduced to identify the provided organizational and operational capabilities or skills of people engaged in Operational and Organizational Roles and assigned to Organizational Units.

No new modelling language constructs are required for any of the remaining modelling phases. The constructs will capture any deviation from the design specification describing the real implementation of the operational system. Similarly decisions on future use of system components will be identified in the decommission phase.

Table E.1 — Example usage of modelling language constructs and their templates

Model phase	Relations to figures E.1 to E.5 and templates		Construct
	Figure E.2: Function View		
1 Domain Identification	Domain: DM-1 Manufacturing Order Processing (Own Enterprise)		Domain
	Domain Boundaries - Domain Inputs: OV-1 / Customer Order, OV-13 / Purchased Parts; - Domain Outputs: OV-11 / Product, OV-8 / Supplier Order;		Enterprise Object Object View
	I/O Origins and Destinations: - DM-X / Customer, DM-Y / Supplier		Event
	Functionalities (Processes): - BP-1 / Administration, BP-2 / Manufacturing		Business Process
	Model design authority: OU-2 / Production Planning		Organizational Unit
2 Concept Definition	DM-1 / Domain Mission, Vision and Values, Operational Policies		Domain
	Decision Function and Level		Decision Centre
3 Requirement Definition	Processes: Administrative BP1 / Manufacturing BP2		Business Process
	Activities: EA-1 / machine, EA-2 / paint, EA-3 / store, EA-4 / assemble		Enterprise Activity
	Events: Shop Floor Order a, Shop Floor Order b		Event
	Activity Inputs (EA4): - Function: Parts, Purchased Parts - Control: Shop Floor Order - Capability: Assembly	Activity Outputs (EA4) - Function Product b - Control: not used	Object View
		- Capability: not used	Capability Operational Role
	Figures E.3: Information View		
	Objects: Part, Product, Market, Order, Report		Enterprise Object
	Views: Part, Purchased Part, Product		Object View
	Figures E.5: Organization View		
	Entities: Enterprise, Department		Organization Unit Decision Centre
4 Design Specification	***Figures E.2: Function View***		
	Processes: Administrative Process BP-1, Manufacturing BP-2		Business Process
	Events: Shop Floor Order a, Shop Floor Order b		Event
	Activity Inputs (EA4): - Function: Parts, Purchased Parts - Control: Shop Floor Order - Resource: Assembly Resource	Activity Outputs (EA4) - Function Product b - Control: Activity Status - Resource: Res. Status	Object View
	Figures E.3: Information View		
	Objects: Part, Product, Market, Order, Report		Enterprise Object
	Views: Part, Purchased Part, Product		Object View
	Figures E.4: Resource View		
	Persons: Supervisor, Operator		Person Profile Operational Role
	Objects: Shop Floor, Assembly Station, Buffer,		Resource
	Objects: Machining Centre		Functional Entity
	Views: Assembly Station, Assembly Station Status		Object View
	Figures E.5: Organization View		
	Entities: Enterprise, Departments: - Shop Floor, Product Development, Production Planning, Purchasing, Sales		Organization Unit
	Members: Supervisor		Organization Role
	Planning Horizon: Month, Week, Day		Decision Centre
5 Implementation Description	Design Specification Model modified by Implementation Deviations from Design Specification. Not part of figures E.2 to E.5		All of the above
6 Domain Operations	The operational use of the Domain Model (released Implementation Description Model). Not part of figures E.2 to E.5		
7 Decommission Definition	Domain Model remodelled for Definition of End-of-Life State of the Domain Components. Not part of figures E.2 to E.5		All of the above

Table E.2 provides a summary of objects in the Order Processing example of Figure E.1. Table E.2 also indicates the identified hierarchies for Enterprise Objects, Resource Objects and Capabilities.

Table E.2 — Example usages of modelling language constructs having multiple instances

Events received

Origin	*ID*	*Name*
DM-Customer	EV-1	Arrival Customer Order
DM-Customer	EV-2	Arrival Customer Payment
DM-Supplier	EV-3	Arrival Purchased Parts
DM Supplier	EV-4	Arrival Invoice
BP-1 Administration	EV-9	Shop Floor Order a
BP-1 Administration	EV-10	Shop Floor Order b

Events sent

ID	*Name*	*Destination*
EV-5	Shipment Product	DM-Customer
EV-6	Sending Invoice	DM-Customer
EV-7	Sending Purchase Order	DM-Supplier
EV-8	Sending Payment	DM-Supplier
EV-11	Completion Product a	BP-1 / Administration
EV-12	Exception4	Exception Handling (TBD)

Enterprise Objects

ID	*Name*
EO-1	Market
- EO-1.1	Customer
- EO-1.1.1	C-Payment
- EO-1.1.2	C-Invoice
- EO-1.2	Supplier
- EO-1.2.1	S-Payment
- EO-1.2.2	S-Invoice
EO-2	Report
EO-3	Status
- EO-3.1	Status Activity
- EO-3.2	Status Operator
- EO-3.3	Status Resource
OR-1	Order
- OR-1.1	Customer Order
- OR-1.2	Purchase Order
- OR-1.3	Shop Floor Order
PR-1	Product
- PR-1.1	Part
– PR-1.2	Drawing
– PR-1.3	BOM
– PR-1.3.1	Material
- PR-1.3.2	Workplan

Table E.2 *(continued)*

Object Views received

Origin	*ID*	*Name*
DM-Customer	OV-1	C-Order
DM-Customer	OV-2	C-Payment
DM-Supplier	OV-3	Purchased Part
DM Supplier	OV-4	Invoice
BP-1 Administration	OV-9	Shop Floor Order a
BP-1 Administration	OV-10	Shop Floor Order b
PR-1.1 Part	OV-12	Painted Part
PR-1.1 Part	OV-13	Purchased Part

Object Views sent

ID	*Name*	*Destination*
OV-5	Product a	DM-Customer
OV-6	Invoice	DM-Customer
OV-7	P-Order	DM-Supplier
OV-8	S-Payment	DM Supplier
OV-11	Product b	DM Customer
OV-14	Activity Information	EO-5 Activity Status
OV-15	Resource Information	EO-6 Resource Status

Organization Units

ID	*Name*
OU-1	Enterprise Operation
OU-2	Production Planning
OU-3	Shop Floor Department
OU-4	ICT Department

Organization Members

ID	*Name*
PPR-1	Supervisor < Person Name >
PPR-2	Operator < Person Name >

Decision Centres

ID	*Name*
DC-1	*Plan Production*

Resource Objects

ID	*Name*
RE-1	Shop Floor
- RE-1.1	Supervisor
- RE-1.2	Operator
- RE-1.3	Computer
– RE-1.3.1	Terminal
- RE-1.4	Assembly Resources
– RE-1.4.1	Robot
– RE-1.4.1.1	Robot Control Program
– RE-1.4.2	Test Station
– RE-1.4.2.1	Test Control Program
– RE-1.4.3	Ass. Fixture
– RE-1.4.4	Ass. Tools
- RE-1.5	Machining Centre
- RE-1.6	Buffer
FE-1	Handling

Table E.2 *(continued)*

Capabilities

ID	*Name*
CA-1	Mechanical Product Assembly
- CA-1.1	Station Set-up and Turn-down
- CA-1.2	Handling
- CA-1.3	Positioning
- CA-1.4	Assembling
CA-2	Mechanical Product Assembly Station
- CA-2.2	Robot
ORR-1	Supervising Shop Floor
OPR-2	Operation of Assembly Station (Skill Profile Operator)

Organizational Role

ID	*Name*
ORR-1	*Supervisor*

Operational Role

ID	*Name*
OPR-2	*Operator*
OPR-3	*Assembly Operator*

Bibliography

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

[2] ISO/IEC 10746 (all parts), *Information technology — Open Distributed Processing*

[3] ISO/IEC 15288:2002, *Systems engineering — System life cycle processes*

[4] ISO/IEC 15414:2006, *Information technology — Open distributed processing — Reference model — Enterprise language*

[5] ISO 15531-1:2004, *Industrial automation systems and integration — Industrial manufacturing management data — Part 1: General overview*

[6] ISO 15704:2000, *Industrial automation systems — Requirements for enterprise-reference architectures and methodologies*

[7] ISO 15704:2000/Amd.1:2005, *Industrial automation systems — Requirements for enterprise-reference architectures and methodologies — Amendment 1: Additional views for user concerns*

[8] ISO/IEC 15909-1, *Software and system engineering — High-level Petri nets — Part 1: Concepts, definitions and graphical notation*

[9] ISO 18629 (all parts), *Industrial automation systems and integration — Process specification language*

[10] ATHENA project FP6 507849, *Deliverable DA1.3.1, Report on Methodology description and guidelines definition*, Public, Version 0.95, March 2005

[11] CIMOSA ASSOCIATION. *CIMOSA — Open System Architecture for CIM: Technical Baseline*. Version 3.2, 1996.[52)]

[12] DOUMEINGTS, G., VALLESPIR, B. and CHEN, D. GRAI Grid Decision Modelling in: Peter Bernus, Kai Mertins, Gunter Schmidt. (eds), *Handbook on Architecture for Information Systems.* (Springer) pp. 313-337, 1998.

[13] REC XML 2000 2006 1006, eXtensible Markup Language (XML), 1.0, 2nd edition, W3C Recommendation October 6, 2000

[14] REC-xmlschema-1-20010502, Schema XML Part 1: Structures — recommendation W3C May 02, 2001

[15] REC-xmlschema-2-20010502, Schema XML Part 2: Datatype — recommendation W3C May 02, 2001

[16] Oxford English Dictionary, © Oxford University Press, 1999

52) Available from www.cimosa.de.

ICS 25.040.40

Price based on 111 pages