ISO/IEC JTC1/SC7/WG17 W17N0129

ITU-T X.911 ISO/IEC 15414

Date: 01-03-03

RM-ODP Enterprise Language FCD with NB Comments Helsinki 2001 Input

ISO/IEC JTC 1/SC 7

Software Engineering

Secretariat: Canada (SCC)

Doc Type:	Final Committee Draft
Title:	Information Technology—Open Distributed Processing— Reference Model—Enterprise Language
	ISO/IEC 15414 | ITU-T Recommendation X.911
Source:	Project Editor
Project:	1.07.77
Status:	FCD with NB Comments, Helsinki 2001 Input
Action:	For Info
Distribution:	SC 7/WG17
Medium:	E
Number of Pages:	132
Version:	303

Address reply to: joaquin@acm.org

ISO/IEC JTC1/SC7 Secretariat
Bell Canada
1050 Beaver Hall Hill 2nd floor
Montréal Québec H2Z 1S4
CANADA
Tel: +1 514 391 8286
Fax: +1 514 870 2246

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.911

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(Final Committee Draft 7/00)

COMMON TEXT DRAFT

Information Technology—
Open Distributed Processing—Reference Model—
Enterprise Language

ITU-T Recommendation X.911

Version 303

CONTENTS

Page

0 Introduction ... 2
0.1 RM-ODP ... 2
0.2 This Recommendation | International Standard ... 3
1 Scope ... 4
2 Normative references ... 5
Identical ITU-T Recommendations | International Standards ... 5
3 Definitions ... 5
3.1 Definitions from ODP standards ... 5
3.1.1 Modelling concept definitions ... 5
3.1.2 Viewpoint language definitions ... 6
3.2 Definitions from ODP standards refined or extended in this standard ... 6
4 Abbreviations ... 6
5 Overview and motivation ... 7
6 Concepts ... 15
6.1 General concepts ... 17
6.2 Role concepts ... 28
6.3 Policy concepts ... 33
6.4 Force concepts ... 36
7 Structuring Rules ... 39
7.1 Overall structure of an enterprise specification ... 39
7.2 Contents of an enterprise specification ... 47
7.3 Community rules ... 49
7.3.1 Specification of a community ... 49
7.3.2 Relationships between communities ... 56
7.4 Enterprise object rules ... 63
7.5 Common community types ... 65
7.5.1 Domain community type ... 66
7.5.2 Federation community type ... 66
7.5.3 Ownership Community Type ... 67
7.6 Lifecycle of a community ... 68
7.6.1 Establishing a community ... 70
7.6.2 Changes in a community ... 71
7.6.3 Terminating a community ... 72
7.7 Objective rules ... 72
7.8 Behaviour rules ... 76
7.8.1 Roles and processes ... 77
7.8.2 Role rules ... 80
7.8.3 Interface roles and interactions between communities ... 83
7.8.4 Enterprise objects and actions ... 85
7.8.5 Process rules ... 89
7.9 Policy rules ... 91
7.9.1 General policy rules ... 93
7.9.2 Obligations, permissions and prohibitions ... 94
7.9.3 Nesting of policy frameworks ... 99
7.9.4 Policy violations ... 99
7.9.5 Organisation of policy ... 100
7.10 Force rules 101
7.10.1 Delegation rules ... 103
7.10.2 Authority rules ... 105

7.10.3 Commitment rules 105
7.10.4 Declaration rules 105
7.10.5 Prescription rules 106
8 Consistency rules 106
8.1 Viewpoint correspondences 106
8.2 Enterprise and information specification correspondences 108
8.2.1 Concepts related by correspondences 108
8.2.2 Required correspondences 110
8.2.3 Required correspondence statements 110
8.3 Enterprise and computational specification correspondences 111
8.3.1 Concepts related by correspondences 111
8.3.2 Required correspondences 112
8.3.3 Required correspondence statements 112
8.4 Enterprise and engineering specification correspondences 113
8.4.1 Concepts related by correspondences 113
8.4.2 Required correspondences 113
8.4.3 Required correspondence statements 114
9 Relations between standards and product development 114
9.1 Compliance, conformance, testing and consistency 114
9.2 Completeness 115
9.3 Scoping statement 116
10 Enterprise Language Compliance 116
11 Conformance and reference points 117
Annex A Overall structure of an enterprise specification 119
Index 124

Foreword

This is the initial final committee draft of ITU | ISO/IEC Common Text for Recommendation X.911 | International Standard 15414: Information Technology—Open Distributed Processing—Reference Model—Enterprise Language. It is the draft output of the May-June 2000 Madrid meeting.

The draft is in the format prescribed for ITU | ISO/IEC Common Text.

The work on this draft Recommendation | International Standard is done by ISO/IEC JTC 1/SC 7/WG 17, originally chartered by SC 21 as a project in SC 21/WG 7, and lately known as SC 33/WG 5 and SC 7/WG 3.

TEMPORARY NOTES The draft includes temporary notes, specified by the working group, for the information of National Bodies, which appear in a smaller font, indented as is this paragraph. These are not part of the text of the draft.

Editor's notes The draft includes editor's notes, which appear in a smaller font, deeply indented as is this paragraph; these are not part of the text of the draft.

Numbering Each line and page of this document is numbered, for the convenience of national bodies commenting on the document.

IMPORTANT: When referring to line and page numbers, use the numbers on the Portable Document Format (PDF) version of this document. The numbers on other versions of the document may change depending on the software or on the printer chosen while viewing the document.

Indexing The project editor requests suggestions for indexing this document.

It is the project editor's faith that a good index is an index prepared by a professional indexer.

Changes to ITU template; these must be changed back:

- Addition of style, Editors Note
- Addition of Keep lines together to Paragraph Line and Page Breaks
- Font

FIN1 G Summary of concerns Inconsistent use of terminology

Problems

Part 2 terminology is not explained enough for readers to follow the discussion

Cobehaviour missing

Implementation objects vs. modeling world at different levels of abstraction still mixed especially when trying to match processes with role

Audiences and usage examples not clear

Force concepts form not a full set, other similar groups of concepts have not been spelled out this way – language is extensible.

Proposal

Suggestions in FIN9, FIN11, FIN14, FIN23, FIN68

ES 30 EAnnex B Examples

Rationale

Having into consideration the intended audience of this document, we feel that some explanatory examples would be of much interest and help to them.

Proposal

Include a second annex with some explanatory text and examples of enterprise specifications. There were some examples in a previous version of this document. We suggest including them, or some modified versions of them.

UK1 Cat G Title

Rationale:

The title of the standard on the cover page of the standard is given as “Information Technology – Open Distributed Processing – Reference Model – Enterprise Viewpoint” although it has been agreed by several meetings of WG 17 and its predecessor bodies that the title is “Information Technology – Open Distributed Processing – Reference Model – Enterprise Language.

Proposal:

The SC 7 Secretariat should be requested (yet again) to note the change of title and to make any consequent changes to the project information.

UK2 Cat G The meaning of the term “enterprise specification”

Rationale:

The terms “enterprise specification” and “enterprise model” are both used, apparently interchangeably, in the current text. If the two terms are considered to be equivalent then only one should be used, if they are considered to be different then their uses should be clearly distinguished.

Proposal:

The terms “enterprise specification” and “specification” should be used in accordance with their use in the RM-ODP Part 3.

The terms “enterprise model” and “model” should only be used to refer to what is expressed by an “enterprise specification” or “specification” respectively.

Editorial changes are proposed in comments below where these are made necessary by this principle.

USA 1 Ethroughout

Rationale

This is a relatively heavyweight method of making minor editorial comments. We understand that the working group does not wish the project editor to make such changes on his own.

Proposal

The USA proposes to take the liberty of suggesting minor editorial changes during the editing meeting.

0 Introduction

The rapid growth of distributed processing has led to the adoption of the Reference Model of Open Distributed Processing (RM-ODP). This Reference Model provides a co-ordinating framework for the standardisation of open distributed processing (ODP). It creates an architecture within which support of distribution, interworking, and portability can be integrated. This architecture provides a framework for the specification of ODP systems.

FIN7 E 0, line 6, page 1 Spelling

Rationale

Missing word.

Proposal

Replace “Open Distributed” with “Open Distributed Processing”.

The Reference Model of Open Distributed is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture.

This Recommendation | International Standard refines and extends the definition of how ODP systems are specified from the enterprise viewpoint, and is intended for the development or use of enterprise specifications of ODP systems.

JP1 E Page 1-2, 0.2, line 9 This Recommendation | International Standard

Rationale

Missing page numbers and a word.

Proposal

Add page numbers and make the sentence readable.

Add page numbers “1” and “2” to each page, and replace “… text ITU-T” by “… text from ITU-T …”, line 9 on the second page.

0.1 RM-ODP

The RM-ODP consists of:

- ITU-T Recommendation X.901 | ISO/IEC 10746-1: **Overview**: which contains a motivational overview of ODP, giving scoping, justification and explanation of key concepts, and an outline of the ODP architecture. It contains explanatory material on how the RM-ODP is to be interpreted and applied by its users, who may include standards writers and architects of ODP systems. It also contains a categorisation of required areas of standardisation expressed in terms of the reference points for conformance identified in ITU-T Recommendation X.903 | ISO/IEC 10746-3. This part is not normative.

- ITU-T Recommendation X.902 | ISO/IEC 10746-2: **Foundations**: which contains the definition of the concepts and analytical framework for normalised description of (arbitrary) distributed processing systems. It introduces the principles of conformance to ODP standards and the way in which they are applied. This is only to a level of detail sufficient to support ITU-T Recommendation X.903 | ISO/IEC 10746-3 and to establish requirements for new specification techniques. This part is normative.

- ITU-T Recommendation X.903 | ISO/IEC 10746-3: **Architecture**: which contains the specification of the required characteristics that qualify distributed processing as open. These are the constraints to which ODP standards must conform. It uses the descriptive techniques from ITU-T Recommendation X.902 | ISO/IEC 10746-2. This part is normative.

- ITU-T Recommendation X.904 | ISO/IEC 10746-4: **Architectural semantics**: which contains a formalisation of the ODP modelling concepts defined in ITU-T Recommendation X.902 | ISO/IEC 10746-2 clauses 8 and 9. The formalisation is achieved by interpreting each concept in terms of the constructs of one or more of the different standardised formal description techniques. This part is normative.

- ITU-T Recommendation X.911 | ISO/IEC 15414: **Enterprise language**: this Recommendation | International Standard.

0.2 This Recommendation | International Standard

ITU-T Recommendation X.903 | ISO/IEC 10746-3 defines a framework for the specification of ODP systems comprising

1) five viewpoints, called enterprise, information, computational, engineering and technology, which provide a basis for the specification of ODP systems;

2) a viewpoint language for each viewpoint, defining concepts and rules for specifying ODP systems from the corresponding viewpoint.

The purpose of this Recommendation | International Standard is to:

--Refine and extend the enterprise language defined in ITU-T Recommendation X.903 |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.

This ITU-T Recommendation X.911 | ISO/IEC IS 15414 uses concepts taken from ITU-T Recommendations X.902 and X.903 | ISO/IEC 10746-2 and 10746-3, and introduces refinements of those concepts, additional viewpoint-specific concepts, and prescriptive rules for enterprise viewpoint specifications. The additional viewpoint-specific concepts are defined using concepts from ITU-T Recommendations X.902 and X.903 | ISO/IEC 10746-2 and 10746-3.

This Recommendation | International Standard contains, for the convenience of the reader, some text taken verbatim from clauses 5 and 10 of ITU-T Recommendation X.903 | ISO/IEC 10746-3: Part 3: Architecture. Such text is marked by a reference like this: [3-5.9], which indicates text taken from part 3, subclause 5.9 of RM-ODP. In the event of any discrepancies in these cases, the text of ITU-T Recommendation X.903 | ISO/IEC 10746-3 is authoritative.

This Recommendation | International Standard also contains some text which is a modification of text ITU-T Recommendation X.903 | ISO/IEC 10746-3: Part 3: Architecture. Such text is marked by a reference like this: [see also 3-5.9]. The modifications are authoritative with respect to the enterprise language.

This Recommendation | International Standard contains these annexes:

Annex A: Overall structure of an enterprise specification

This annex is not normative.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY—OPEN DISTRIBUTED PROCESSING—REFERENCE MODEL—ENTERPRISE LANGUAGE

1 Scope

This Recommendation | International Standard provides:

a) a language (the enterprise language) comprising concepts, structures, and rules for developing, representing, and reasoning about a specification of an ODP system from the enterprise viewpoint (as defined in ITU-T Recommendation X.903 |ISO/IEC 10746-3);

b) rules which establish correspondences between the enterprise language and the other viewpoint languages (defined in ITU-T Recommendation X.903 |ISO/IEC 10746-3) to ensure the overall consistency of a specification.

The language is specified to a level of detail sufficient to enable the determination of the compliance of any ODP modelling language to this Recommendation | International Standard and to establish requirements for new specification techniques.

JP2 TL Page 3, 1, forth paragraph, lines 15 Scope

Rationale

Whether modelling language is “ODP modelling language” or just a “modelling language” should be dependent on the results of conformance testing (if we do have it). If the modelling language is called ODP modelling language, then it should be compliant, and the text seems inappropriate.

Also what is important here is not a modelling language but models created with modelling languages.

Proposal

Remove reference to ODP.

Remove “ODP” from “ … compliance of any ODP modelling language …” and make it more general.

This Recommendation | International Standard is a refinement and extension of ITU-T Recommendation X.903 | ISO/IEC 10746-3, clauses 5 and 10, but does not replace them.

This standard is intended for use in preparing enterprise viewpoint specifications of ODP systems, and in developing notations and tools to support such specifications.

As specified in clause 5 of ITU-T Recommendation X.903 | ISO/IEC 10746-3, an enterprise viewpoint specification defines the purpose, scope and policies of an ODP system. [3-5.0]

2 Normative references

The following ITU-T Recommendations | International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations | International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent editions of the Recommendations | International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The ITU-T Secretariat maintains a list of the currently valid ITU-T Recommendations.

FIN8 E 2, page 3, line25 Spelling

Proposal

Replace “con tain” with “contain”.

Identical ITU-T Recommendations | International Standards

– ITU-T Recommendation X.902 (1995) | ISO/IEC 10746-2: 1994, *Information technology – Open Distributed Processing – Reference Model – Foundations*

– ITU-T Recommendation X.903 (1995) | ISO/IEC 10746-3: 1994, *Information technology – Open Distributed Processing – Reference Model – Architecture*

– ITU-T Recommendation X.904 (1997) | ISO/IEC 10746-4: 1997, *Information technology – Open Distributed Processing – Reference Model – Architectural semantics*

3 Definitions

3.1 Definitions from ODP standards

3.1.1 Modelling concept definitions

This Recommendation | International Standard makes use of the following terms as defined in ITU-T Recommendation X.902 | ISO/IEC 10746-2

-- action;

-- behaviour (of an object) ;

-- composite object;

-- composition;

-- configuration (of objects) ;

-- conformance;

-- conformance point;

-- contract;

-- <X> domain;

-- entity;

-- environment contract;

-- environment (of an object) ;

-- epoch;

-- establishing behaviour;

-- incremental modification;

-- instance (of a type) ;

-- instantiation (of an <X> template) ;

-- interface;

-- internal action;

-- interworking reference point;

-- invariant;

-- liaison;

-- location in space;
-- location in time;
-- object;
-- obligation;
-- ODP standards;
-- ODP system;
-- perceptual reference point;
-- permission;
-- policy;
-- postcondition;
-- precondition;
-- programmatic reference point;
-- prohibition;
-- proposition;
-- quality of service;
-- reference point;
-- refinement;
-- responding object;
-- role;
-- state (of an object) ;
-- subtype;
-- system;
-- <X> template;
-- terminating behaviour;
-- type (of an <X>);
-- viewpoint (on a system).

3.1.2 Viewpoint language definitions

This Recommendation | International Standard makes use of the following terms as defined in ITU-T Recommendation X.903 | ISO/IEC 10746-3

-- community;
-- computational interface;
-- computational viewpoint;
-- dynamic schema;
-- engineering viewpoint;
-- enterprise viewpoint;
-- <X> federation
-- information viewpoint;
-- invariant schema;
-- <viewpoint> language;
-- static schema;
-- technology viewpoint.
.

3.2 Definitions from ODP standards refined or extended in this standard

This Recommendation | International Standard refines or extends the definitions of the following terms originally defined in ITU-T X.902 | ISO/IEC 10746-2 (the refined or extended definitions are in clause 6):

-- policy

4 Abbreviations

ODP open distributed processing

RM-ODP Reference Model of Open Distributed Processing
(ITU-T Recommendations X.901 to X.904 | ISO/IEC IS 10746)

5 Overview and motivation

FIN9 TH clause 5 induced rewrite of clause 5

Rationale

Collected together the comments on the audience of this standard, on the exploitation scenarios for this standard, and the clarity and spelling of clause 5 (FIN9-FIN14) induce the following rewrite of clause 5.

Proposal

5. Motivation and overview

The purpose of this Recommendation | International Standard is to provide 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. Such an enterprise specification forms a part of the specification of and ODP system in terms of the set of viewpoints defined by ITU-T Recommendation X.903 | ISO/IEC 10746-3.

The primary audience for the document consists of those who prepare and use such specifications. The motivation for a standard enterprise language is to support standardised techniques for specification. This improves communication and helps create consistent overall specifications.

The preparation of specifications often falls into the category referred to as analysis of requirement specification. There are many approaches used for understanding, agreeing and specifying systems in the context of the organisations of which they form a part. The approaches can provide useful insights into both the organisation under consideration and the requirements for systems to support it, but they generally lack the rigour, consistency and completeness needed for thorough specification. The audiences of the specifications also vary. For agreement between the potential clients of an ODP system and the provider of that system, there may be needed to have different presentations of the same system - one in terms understood by clients, and one in terms directly related to system realization.

The use of enterprise specifications can be wider than the early phases of software engineering process. Current trend is to integrate existing systems into global networks, where the functionality of interest spans multiple organisations. The enterprise language provides means to specify the joint agreement of common behaviour of the ODP systems within and between these organisations. The enterprise specification can also be used at other phases of the system life-cycle. The specification can for example be used at system run-time to control agreements between the system and its users, and to establish new agreements according to the same contact structure. Enterprise viewpoint specifications may contain rules for inter-organisational behaviour.

This standard also provides a framework for development of software engineering methodologies and tools exploiting ODP viewpoint languages, and a set of concepts for development of denotational languages for ODP enterprise language itself. For these purposes this standard provides rules for the information contents of the specification and the grouping of that information. Further requirements on the relationships between enterprise language concepts and their mapping to concepts in other viewpoints are specific to the methodologies, tools or denotational languages to be developed.

An enterprise specification defines the scope, purpose and policy framework of an ODP system. The specification gives a statement of conformance for system implementations. The system purpose

becomes defined by the specified behaviour of the system; policies capture further restriction of the behaviour between the system and its environment or within the system itself. Policy is a set of rules related to a particular purpose. The expressed obligations, permissions and prohibitions are in the context of communities naturally related to the business decisions of the system owners.

An ODP system may span multiple domains and need not to be owned by a single party. The system can also be divided to independently specified and independently working subsystems. In such a case, an essential part of the enterprise specification is to capture the collective behaviour of the full system.

The enterprise specification comprises of a set of community specifications. The division to communities can be made based on functional grouping, control domains or ownership reasons. For each enterprise specification there should be a root community specification in which the scope of the system becomes explicated. This community may represent the ODP system as a single object or if the division for example to multiple owners is important to a set of ODP system objects, each of which become further refined by other community specifications.

A community specification captures the requirements for objects in terms of roles and interactions between these roles. The community specification also captures policy statements further stating what rules guide the system in situations where multiple behaviour patters are possible. Policies can be separately set for each case in which the same specification is otherwise used.

The community that conforms to the community specification is comprised of enterprise objects. The enterprise specification may include the definition of these objects or trust on supportive mechanisms to introduce those objects into the communities during the life-time of the communities.

The community specification contains rules for assigning enterprise objects to roles. These rules reflect the way in which enterprise objects are introduced and the way in which binding of objects to roles can be implemented in the factual system environment.

Some basic concepts used in this standard are defined in RM ODP Part 2 or 3. The enterprise language definitions use them in the following contexts.

A community specification captures the contract between objects within the specified community. For a community contract the essential aspects of the contracts are

- the behaviour made possible by the roles involved;
- policies governing the behaviour of the community;
- rules for changing the structure of the community (roles, policies); and
- rules for deciding when the community contract is violated and what actions are taken then.

A role is an identifier for a behaviour. A role is used to project a behaviour that needs to be assigned to a single enterprise object. A behaviour is generally defined to be a collection of actions with constraints on their occurrences. A behaviour is always associated with an object or with a composition of objects. For communities, the interesting behaviour is formed by the interactions between the community and its environment and between the objects fulfilling roles in the community. A role thus identifies a projection of the community's collective behaviour specification.

A process is an alternative projection of the community's collective behaviour; an action belongs to a process if it supports a shared goal independent on the performer of that action.

Role, contract and policy concepts are used as directly used as defined in ITU-T Recommendation X.902 | ISO/IEC 10746-2, other concepts are defined in this Recommendation | Standard.

FIN14 TH G, 5 Relationship between Part 2 and EL

Rationale

Part 2 concepts are general and thus difficult to generally apply to enterprise language. The application of the concepts need to be refined for each specific context, thus the readers should be given some guidance for interpretation.

Proposal

Add role, behaviour, action, activity, contract, policy, etc introductions and interpretations to clause 5, which currently works only as a motivation section. See FIN9 for proposed text.

FIN50 E7.8.1, 5 inappropriate text style

Rationale

Descriptive text, not normative rules. Not suitable for clause 7.

Proposal

Move 7.8.1 to 5.

Add to 5 the idea that behaviour refers to collective behaviour unless otherwise stated: an object can always have individual behaviour but the object can also be a composite object or represent a community.

The behaviour can be split into specifiable segments by projecting it into roles or processes. The roles and processes both can be subdivided into steps, which bridges between these two projections together. Policies need to have the granularity of step to be usable for both specification styles.

Possibly add notes for 6.1.5, 6.1.6 for the relationship.

TEMPORARY NOTE – Canada has made this comment: “It is not clear if the target audience of this document is. This can affect the style of the document.

“If the target audience is those people who design languages for such type of specification, then the style has to be quite formal, and the normative part of the standard rigourous enough to enable conformance verification.

“If the target audience is those who prepare or use such specification, then the nature of the document is more like a guideline, and the style is less formal.

“The target audience of the document should be clearly stated in clause 1, and the requirements from this target audience stated in clause 5.”

The purpose of this Recommendation | International Standard is to provide 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. Such an enterprise specification forms part of the specification of an ODP system in terms of the set of viewpoints defined by ITU-T Recommendation X.903 |ISO/IEC 10746-3. The primary audience for the document consists of those who prepare and use such specifications. It also provides a rigorous framework for the development of supporting methods and tools.

FIN10 TH 1,5, 8 Audience of the document

Rationale

In clause 5, lines 26-33 of page 5, Canada expects to hear about the intended audience of this document. The Enterprise language should address both audiences, and this statement is already present in Clause 1. In addition, clause 5 should contain clearer statements on how these audiences should use the language in various situations. The list of situations should be expanded. The use as a basis for tool and language construction is also reflected to clause 8, where requirements should be set for tool providers to nail down some of the correspondence statements now left free for the specifiers.

Proposal

Proposed changed text for clause 5, lines 38-39: Replace the sentence "It also provides a rigorous framework for the development of supporting methods and tools" with a new paragraph:

This standard also provides a framework for development of software engineering methodologies and tools exploiting ODP viewpoint languages, and a set of concepts for development of denotational languages for ODP enterprise language itself. For these purposes this standard provides rules for the information contents of the specification and the grouping of that information. Further requirements on the relationships between enterprise language concepts and their mappings to concepts in other viewpoints are specific to the methodologies, tools or denotational languages to be developed.

ES 1 E Page 5, line 34 Use of "Rec | Int'l Standard"

Rationale

The editor's note in page 5, lines 40-43 points out the fact that the use of the term "Recommendation | International Standard" at this point contravenes the ISO rules for presentation.

Proposal

We suggest using the term "Enterprise language" in that section, instead of "Recommendation | International Standard". Therefore, we propose to change line 34 to:

The purpose of the Enterprise Language is to provide a common language (set of terms and structuring ...

USA 5 H Page 5, 5, line 34 Overview and motivation

Rationale

Compliance with the rules for presentation of ITU-T | ISO/IEC common text.

The USA made this same proposal in its ballot comments on the previous committee draft. The working group gave as its reason for rejecting the proposal: "it's not a Reference Model." The Enterprise Language is a part of the Open Distributed Processing Reference Model, so we again propose this change.

Proposal

Change: this Recommendation | International Standard

To:

this Reference Model

USA 6 H Page 5, 5, line 34 Overview and motivation

Rationale

Compliance with the rules for presentation of ITU-T | ISO/IEC common text.

This was proposed in earlier ballot comments. The working group gave as its reason for rejecting the proposal: "the WG did not feel that this is a good name." This document prescribes and specifies the Enterprise Language. We urge the working group to not again reject this proposal without making some change to comply with the common text rules.

Proposal

Change: this Recommendation | International Standard

To:

this prescription of the ODP enterprise language

or

this specification of the ODP enterprise language

Editor's Note – The use of the term, 'Recommendation | International Standard,' at this point in the document contravenes Clause 1 of Rules for presentation of ITU | ISO/IEC Common Text, which prescribes that "a term which is descriptive of the nature of the common text should be used when the document refers to itself" in clauses after the Scope clause.

Accordingly, a descriptive term for this document must be chosen. Possible terms include 'reference model,' 'prescription of the ODP enterprise language' and 'language.' (This Recommendation | International Standard is: Information Technology—Open Distributed Processing—Reference Model—Enterprise Language.

There are many approaches used for understanding, agreeing and specifying systems in the context of the organizations of which they form a part. Many of these approaches fall into the categories often referred to as analysis or requirements specification. They can provide useful insights into both the organization under consideration and the requirements for systems to support it, but they generally lack the rigour, consistency and completeness needed for thorough specification. It is a key objective of this Recommendation | International Standard to provide a way of relating the commonly used concepts and underlying principles of such approaches to the modelling framework of the RM-ODP.

FIN11 TL 5 Usage examples

Rationale

Needs more concrete understanding on the situations in which enterprise language can be applied.

Proposal

Proposed changed text at line 5, page 6, first sentence after note:

When preparing a specification, there are many approaches used for understanding, …

Proposed new paragraph between lines 34 and 35, page 6:

Current trend is to integrate existing systems into global networks, where the functionality of interest spans multiple organizations. The enterprise language provides means to specify the joing agreement of common behaviour of the ODP systems within and between these organizations.

The enterprise language provides the terms and structuring rules to specify the purpose, scope and policies for an ODP system in a manner that is meaningful for the stakeholders for that system, including the owners, the users and the developers. An enterprise specification describes the behaviour of the system within the environment with which it interacts. Such an environment can be a technical environment (e.g., the software and hardware environment of a service component) or a social or business organisation (e.g., a group of co-operating companies, a particular service inside a company).

ES 2 E Page 6, lines 12-14 Stakeholders of the system

Rationale

Owners, users and developers are mentioned as stakeholders for a system. Another important group of stakeholders is *maintainers*, which are responsible for incorporating changes to the system in order to fix bugs or to update the system according to its natural evolution and future upgrades.

In this sense, an Enterprise Specification should, apart from "describing the behaviour of the system within the environment with which it interacts" [page 6, lines 14-15], describe also the policies that rule the evolution of the system.

Proposal

Change lines 12-14 in paragraph 3, page 6, with:

The enterprise language provides the terms and structuring rules to specify the purpose, scope and policies for an ODP system in a manner that is meaningful for the stakeholders for that system, including the owners, the users, the developers, and the maintainers.

And add the following sentence to the end of paragraph 5, lines 20-25: "Policies about potential changes in the system that may rule its future evolution may also be considered". This leaves the paragraph as follows:

An enterprise specification of an ODP system is an abstraction of the system and a larger environment in which the ODP system exists, describing those aspects that are relevant to specifying what the system is expected to do in the context of purpose, scope and policies of its environment (technical, organisational). It describes the behaviour assumed by those who interact with the ODP system. It explicitly includes those aspects of the environment that influence the behaviour of the ODP system – environmental constraints are captured as well as usage and management rules. Policies about potential changes in the system that may rule its future evolution may also be considered.

USA 4 TH Page 5, 5, lines 14-15, 21-22 Overview and motivation

Rationale

The overview and motivation are formulated mostly in terms of the behaviour of an ODP system. At the same time, Part 2 states that state and behaviour are dual concepts. The choice of a concept is often determined by the ease of understanding by human readers as well as by the elegance of the specification. Since invariant properties of communities often determine the behavioural properties of these communities, and since the invariant properties (such as constraints) are often more stable than the volatile behavioural properties, it makes sense to make a more explicit emphasis on the invariant properties.

Note. This rationale and proposal apply also to page 28, lines 3-7 (9.3 Scoping statement).

Proposal

Change: An enterprise specification describes the behaviour of the system within the environment...

To:

An enterprise specification described the structural and behavioural properties of the system within its environment...

Change: what the system is expected to do

To:

what the system is expected to be and to do

The enterprise language defines the concepts necessary to represent the behaviour expected of an ODP system. It defines structuring rules for using those concepts to produce an enterprise specification.

An enterprise specification of an ODP system is an abstraction of the system and a larger environment in which the ODP system exists, describing those aspects that are relevant to specifying what the system is expected to do in the context of purpose, scope and policies of its environment (technical, organisational). It describes the behaviour assumed by those who interact with the ODP system. It explicitly includes those aspects of the environment that influence the behaviour of the ODP system – environmental constraints are captured as well as usage and management rules.

JP3 TL Page 6, 5, third paragraph from bottom, lines 22Overview and motivation

Rationale

Purpose, scope and policies are for ODP systems, not for environment.

Proposal

Rewording is necessary.

Replace the current text with the following.

"… expected to do in the context of purpose, scope, and policies."

UK4 Cat E 5 Overview and motivation

Proposal:

a) Page 6, line 20. Replace "a larger environment" by "the environment".

b) Page 6, line 22.

i) Insert "the" before "purpose, scope…".

ii) Replace "(technical, organisational)" by "(technical, social or business)" - aligning with line 16.

An important objective of an enterprise specification is to support an agreement (for example, as part of the contract for the supply of a system) between the potential clients of an ODP system and the provider of that system. Both parties should be able to write, read and discuss such a specification, the clients to be sure of the expected behaviour of the system that they will get, and the provider to be clear about the behaviour to be realised by the system being provided. Thus, two types of presentation of the enterprise specification may need to be considered for the same system. One presentation may need to provide a view of the specification in

terms that are understood by the clients. A second presentation may be needed to present the specification in terms that more directly relate to its realisation. Both types of presentation address enterprise considerations as they concern the system.

FIN12 E5, page 6, line 30, lines 33-34 Clarity

Proposal

Replace “Thus, two” with “Several”.

Remove last sentence of the paragraph.

FIN11b TL 5 Usage examples

Rationale

Needs more concrete understanding on the situations in which enterprise language can be applied.

Proposal

…

Proposed new paragraph between lines 34 and 35, page 6:

Current trend is to integrate existing systems into global networks, where the functionality of interest spans multiple organizations. The enterprise language provides means to specify the joing agreement of common behaviour of the ODP systems within and between these organizations.

AFNOR 3 TL Page 6, Clause 5, Paragraph 5, lines 26-34

Rationale

These lines deal with method although the RM-ODP standards are free from method.

Proposal

Put them as a note.

JP4 TL Page 6, 5, third paragraph from bottom, lines 30-34 Overview and motivation

Rationale

No description about “two types of presentation” is given later in the document. The text may not be normative.

Proposal

Remove the text or add reference(s) to the places that describe “two types of presentation” rules.

Remove the text “Thus, two types of presentation … “ or add reference to corresponding clause(s).

The motivation for a standard enterprise language is to support standardized techniques for specification. This improves communication and helps create consistent overall specifications. One way of using the enterprise viewpoint is to discover enterprise objects by focusing on purpose, scope and policies of the system, and then define other viewpoint models, e.g., computational objects, based on the correspondence with enterprise objects.

FIN13 E5, page 6, lines 35-39 Clarity

Proposal

Replace the last paragraph with:

The enterprise specifications can also be used at other phases of the system life-cycle. The specification can for example be used at system run-time to control agreements between system and its users, and to establish new agreements according to the same contract structure. Enterprise viewpoint specifications may contain rules for inter-organizational behaviour.

AFNOR 4 TL Page 6, Clause 5, Paragraph 6, lines 35-39

Rationale

These lines deal with method although the RM-ODP standards are free from method.

Proposal

Delete these lines or put them as a note

UK5 Cat TL 5 Overview and motivation

Rational:

Page 6, line 36-39. The last sentence gives a specific view of the use of an enterprise specification that is out of place here.

Proposal:

Page 6, line 36-39. Delete the last sentence

6 Concepts

UK6 Cat E The ordering and grouping of definitions in clause 6

Rationale:

The set of definitions has changed significantly since the order and grouping was set up.

Proposal:

a) The categorisation of concepts and the ordering of the categories should reflect the subclauses and the order of subclauses in clause 7. The following categories are proposed:

6.1 Community concepts

6.2 Behaviour concepts

6.3 Policy concepts

6.4 Responsibility concepts

b) The following is proposed as the grouping and ordering of the concepts in these categories:

6.1 Community concepts

6.1.1 Objectiv

6.1.2 Purpose (of a system)
6.1.3 Scope (of a system)
6.1.4 S-community
6.1.5 C-object
6.2 Behaviour concepts
6.2.1 Actor (with respect to an action)
6.2.2 Artefact (with respect to an action)
6.2.3 Resource
6.2.4 Actor role (with respect to a community)
6.2.5 Artefact role (with respect to a community)
6.2.6 Resource role (with respect to a community)
6.2.7 Interface role (with respect to a community)
6.2.8 Process
6.2.9 Step
6.3 Policy concepts
6.3.1 Policy
6.3.2 Authorisation
6.3.3 Violation
6.4 Responsibility concepts
6.4.1 Purposeful selection
6.4.2 Act
6.4.3 Party
6.4.4 Machine
6.4.5 Commitment
6.4.6 Declaration
6.4.7 Delegation
6.4.8 Evaluation
6.4.9 Prescription
6.4.10 Agent
6.4.11 Principal
6.4.12 Contracting party (with respect to a contract)
6.4.13 Owner (of an <X>)

USA 2 G Action

Rationale

This is parallel with:

2-8.1 Object: A model of an entity.

It is consistent with the rest of the text of Parts 2 and 3.

This will help to clarify the distinction between the elements of the universe of discourse and the elements of a specification, model, or abstraction (i.e., simplified model).

In the Madrid meeting dispositions, this was marked Category 4, Noted. In making these comments, we have interpreted this disposition as meaning that the working group agrees with this reading of Part 2. If we are mistaken, we apologize for not having had this clarified earlier.

Proposal

Proceed, in our work, as if the Part 2 definition of ‘action’ were:

2-8.3 Action: A model of something which happens.

The concepts of the enterprise language defined in this Recommendation | International Standard comprise:

--the concepts identified in 3.1.1 and 3.1.2 as they are defined in ITU-T X.902 | ISO/IEC 10746-2 and in ITU-T X.903 | ISO/IEC 10746-3;

--the concepts defined in this clause.

The concepts defined in this clause include both new concepts and refinements of concepts from ITU-T X.902 | ISO/IEC 10746-2 and ITU-T X.903 | ISO/IEC 10746-3. The grouping into subclauses and the headings of the subclauses of this clause are not normative.

6.1 General concepts

6.1.1 Purpose (of a system): The practical advantage or intended effect of the system.

AFNOR 5 TH Page 7, Clause 6.1.1

Rationale

First, although this definition is very vague, it must be in this standard because purpose is mentioned in part 3. Second, we think that a purpose can be applied to any enterprise object.

Proposal

6.1.1 Purpose (of an object): The practical advantage or intended effect of an object.

NOTE

1 – Purpose is specially useful for the system when it is seen as an object

6.1.2 Objective: Statements of preference about possible future states, which influence the choices within some behaviour.

FIN15 TL 6.1.2 Objective

Rationale

The system objective is not a syntactical entity. Therefore, the word sentences is extra. The representations of the objective become visible in behaviour, contracts and policies as described in FIN9.

Proposal

Replace the definition by

“Preferences about possible future states, which influence the choices within some behaviour”.

Add note:

3 – Statements of objective may be an explicit part of the specification depending on the denotation and methodology selected. The objective of the system may also be indirectly present for example in policies governing the system and its behaviour.

Remove note 2.

AFNOR 6 TH Page 7, Clause 6.1.2

Rationale

The fact that, when an object has an objective this influence its choices within some behaviour, is a structuring rule.

Proposal

Change the definition to :

6.1.2 Objective: Statements of preference about possible future states.

Move the rest of the definition to the clause 7.7 Objective rules.

Remove the notes.

JP5 TL Page 7, 6.1.2, line 6 Objective

Rationale

Objective is something to be met or achieved, and maybe stated as statements. Statement is just a form of expressing objective and thus not an essential part of the definition.

Proposal

Remove unnecessary part from the text.

Replace the current text with the following.

6.1.2 Objective: Preference about possible future states, which influence the choices within some behaviour.

USA 7 TL Page 7, 6.1.2, lines 6-7 Objective

Rationale

This is briefer, makes the point, and avoids questions about choices in other contexts than a within some behaviour.

Proposal

Change:

6.1.2 Objective: ~~Statements of~~ Preferences about possible future states, which influence the choices within some behaviour.

NOTES

1 – Some objectives are ongoing, others are achieved once met.

2 – The choices influenced by an objective are not necessarily choices of the holder of the objective.

Editor's Note - Could it be that an objective is not a statement? That it is a statement of an objective that is a statement? If so, do we want: Objective: Preferences about possible future states which influence the choices within some behaviour.

6.1.3 Scope (of a system): The behaviour that system is expected to exhibit.

AFNOR 7 TH Page 7, Clause 6.1.3

Rationale

We are quite disappointed with this definition. For us the system is **expected** to conform to all the specification, not only the scope. We think that the scope is a kind of abstraction of the behaviour of the system.

Proposal

Change the definition to :

Clause 6.1.3 Scope (of a system): An abstraction of the behaviour of the system.

TEMPORARY NOTE – The working group invites National Body consideration whether there is a need for a concept that expresses what the system is capable of doing. (Such a concept would distinguish the delivered behaviour from the expected behaviour.)

FIN16 A 6.1.3 Scope, answer to question raised

Rationale

We feel no need for additional concepts here. When there is a need to express the actual capabilities of a system, the terms capability and system behaviour are available.

ES 3 E Page 7, Temporary Note, lines 15-17 "Delivered Behaviour"

Rationale

We agree with the text in the temporary note, and feel that a concept is indeed needed to differentiate between the behaviour that a system is expected to exhibit (its "**scope**"), and the behaviour that the implemented system actually delivers (its "**delivered behaviour**"). An example of the utility of this concept becomes visible when trying to specify the possible deviations that the implemented system may have with respect to its expected behaviour, in order to decide the acceptance or rejection of the system (some deviations may be acceptable, and even subject to specification).

Proposal

Include a new concept after 6.1.3:

6.1.4. Delivered behaviour (of a system): the behaviour that the actual system is capable of doing (as opposed to its specified expected behaviour, i.e., its scope).

JP6 TL Page 7, 6.1.3 Temporary Note, line 15 Scope

Rationale

The delivered behaviour of system and the expected behaviour of system could be distinguished by saying "behaviour of system" and "scope of system.".

Proposal

Remove or modify the note.

Remove the current temporary note.

USA 8 TH Page 7, 6.1.3, line 14 Scope

Rationale

Responding to the invitation to comment for the need for an additional concept: The distinction between expected behaviour and delivered behaviour is important. However, this subject is well covered by the ODP approach to conformance. [2-15] No additional concept is needed.

Proposal

?

6.1.4 Purposeful selection: An action selecting one of a set of possible behaviours where the behaviour initiated is the one that best suits an objective, as determined by some selection criteria.

NOTES:

1 – A purposeful selection may take place as an internal action of an enterprise object or by the collective behaviour of several enterprise objects.

2 – The set of possible behaviours may not be predetermined and the selection criteria may not be explicit. This may, in particular, be the case when the action of selection involves parties.

FIN18 TH 6.1.4 Purposeful selection

Rationale

Policies as means of controlling alternative behaviours is equally powerful, already existing mechanism.

Proposal

Remove 6.1.4.

UK3 Cat G, TH Clauses 6.1.4, 6.4 and 7.10 Purposeful selection and Force concepts

Rationale:

1. The UK has spent some time reviewing the force concepts and has found it difficult to get a clear picture of their nature and purpose, and how they would be used in an enterprise specification. Issues of concern and requirements for clarification are:

a) The force concepts appear to be concerned with expressing patterns of responsibilities that apply to parties and how these are related to the applicable legal framework. However, the significance of the term “force” is not clear and no direction is provided for incorporating the concepts in a specification.

b) Purposeful selection appears to be concerned with a choice of behaviours to meet an objective. The notion of a choice of actions appears to be closely related to what is said about policy in 7.9, but this relationship is not discussed.

c) There is a need to clarify a number of specific issues related to the concept of “act”:

- Why is purposeful selection limited to “acts”? What is significant about “acts” that they have to be the result of purposeful selection?

- What is the distinction between acts and actions, and the relation between acts and responsibility? The meaning of the term “act” (6.4.1) is only stated indirectly in the definition, but an “act” appears to be a choice between actions (a purposeful selection) that creates some obligation - when made by an agent it commits the principal to the choice;

- How does a specification, for an agent, identify which actions are acts on behalf of which principal?

- What are the delegation rules for acts?

d) The introduction of the declaration and prescription concepts does not seem justified:

declaration talks about changes in the environment of an object as a result of internal actions of an object - internal actions of an object cannot, by definition, affect its environment;

prescription seems to be a special case of commitment - and to be covered by or, at least, be closely related to policy rules in 7.9.

e) There is a need to clarify the use of the term "responsibility". Two "levels" of use can be distinguished, namely:

use of the term to indicate that one enterprise object relies upon by another to provide some service;

use of the term to indicate the handling of exception conditions either by direct reference outside the specification or by a choice between alternative actions where the mechanism for choice is outside the specification.

The second use is the one that relates to party - in other words, responsibility is related to the passing of responsibility for the handling of exception conditions.

Thus, an enterprise specification may simply say that exception conditions are handled by actions that are not part of the specification, or it may specify a choice of actions to be taken, with the mechanism of choice unspecified and leaving the choice to be made by a party.

f) According to page 22, lines 16 and 17, responsibility cannot be delegated. It follows that, in terms of responsibility, an agent has a prescribed behaviour and exceptions must be referred to the principal with an eventual reference outside a specification or with a choice of actions to be made by a principal that is a party.

2. The UK believes that the following is a valid interpretation of the concepts:

a) 6.4.says that the point of the force concepts is "to model changes in that part of the universe of discourse modelled by the environment of the ODP system, including such changes when caused by changes in the ODP system itself." These changes are the result of "acts". "Acts" can only be carried out by parties or the agents of parties - since responsibility for an "act" can only rest with a party.

b) However, the purpose of the enterprise specification is to describe the activities of the environment of the ODP system in which the ODP system participates - and the purpose of the activities described is to bring about changes in the state of that environment.

c) It seems to follow from (b) that all actions in an enterprise specification are "acts" (shades of "performative actions";-)) and, hence, there must be an identified principal for every action (act) of the ODP system and, thus, it is the agent of one or principals (parties).

d) If this is valid, then there is, clearly, a need to talk about act, party, agent, delegation and principal in order to be able to specify responsibility and delegation within an enterprise specification. This should be related to the clause on behaviour.

e) There does not seem to be a need to classify acts beyond delegation i.e. the specification of commitment, declaration, evaluation and prescription is not required.

In this context it is unclear how purposeful selection fits in. Since 7.7 says that everything about the enterprise specification is expected to be driven by the objective, what is special about purposeful selection?

Proposal:

The issues above require clarification, however the UK does not believe that it has a sufficient understanding of the text to provide proposals.

It is essential for text to be provided or developed to cover these issues in order for the document to progress to FDIS. In view of the urgency of progressing the document, the UK believes that the concepts should be deleted if such text is not available and cannot be developed during the Editing Meeting.

6.1.5 Process: A collection of steps taking place in a prescribed manner and leading to the accomplishment of some result.

NOTES

1 – A process may have multiple starting points.

FIN19 TH 6.1.5 Process

Rationale

Continuous processes are denied.

We don't understand the need of multiple starting points. How is starting point defined?

What is a result?

Is there really difference between activity and process?

Is a process a partially ordered sequence of parallel activitites of enterprise objects.

Could we use the term workflow instead?

Clause 7.8 is more clear in this respect and material from there should be moved here as a definition.

Proposal

Replace the current definition with the following.

Process: Composite behaviour of the community leading to a preferred state in the community. Process can be expressed as a partially ordered sequence of steps.

1 - Process is an activity with objective.

2 - Activities can be refined and thus repetitive behaviour patterns can be modelled.

Note – The process specification is independent of the identity of performing enterprise object.

Note - The process specification is abstract, and clearly differs from a computational description of the cobehaviour between actors. The goal of process specification is to give a partial order for actions within the community, not to fully specify the behaviour.

Note – A process specification can represent a workflow description.

Note – Role is thus an identifier for a part of a composite behaviour. Role can be expressed as a collection of steps it participates and its internal behaviour.

ES 4 TL Page 7, 6.1.5, lines 25-26 Process

Rationale

A process does not need to produce anything (e.g., a result). In general, processes should be aimed at objectives.

Proposal

Change the definition of process to:

6.1.5 Process: A collection of steps taking place in a prescribed manner and leading to the achievement of some objective.

JP7 TL Page 7, 6.1.5, NOTE 1 line 28 Process

Rationale

NOTE 1 only states about starting points. End points may also be added.

Proposal

Add statement about end points into the current text.

Replace the current text with the following.

1 – A process may have multiple starting points and multiple end points.

2 – The activity structure concepts provided in subclause 13.1 of ITU-T Recommendation X.902 | ISO/IEC 10746-2 may be used, after substitution of 'step' for 'action' and 'process' for 'activity,' to specify the structure of a process.

3 – An enterprise specification may define types of processes and may define process templates.

6.1.6 Step: An abstraction of an action, used in a process, that may hide objects with which that action is associated.

FIN20 TH 6.1.6 Step

Rationale

Misreading of part 2…. 8.3 does not require an identified object to be associated.

Proposal

Step: An action associated with one or more roles.

1 – An action is a modeling concept that is not reserved to a single occurrence of an action. However, the term action is for short also used for action occurrences.

2 – As the granularity of actions is a design choice, the actions can be further refined as more detailed behaviour.

USA 9 TH Page 7, 6.1.6, lines 32-33 Step

Rationale

This has the same meaning, but avoids the new usage by which an object 'associated' with an action.

Proposal

Change:

6.1.6 Step: An abstraction of an action, used in a process, that may hide <u>the</u> objects ~~with which that action is associated~~ <u>that participate in that action</u>.

6.1.7 S-community: A community in which an ODP system is represented as a single enterprise object interacting with its environment

FIN21 TH 6.1.7 S-community

Rationale

The need for a community with a special role is not properly revealed by the text, thus also the intended use of the concept is left unclear. The term has been under debate quite a while.

Proposal

Proposed replacement text:

Root community specification. A community specification in which an ODP system can fulfil a role. The role specification thus expresses how the ODP system is required to interact with its environment.

Add notes:

1 - The root community is a representation of the scope of a system.

2 - For example, in cases where federation needs to be modeled, there may be more than one ODP systems simultaneously fulfilling roles in the root community.

3 - The ODP system is represented as a single enterprise object, possibly as a community object.

4 – For expressing an environment contract (for example for reuse purposes and runtime binding across ODP systems) it is preferred that the root community specification includes a scoping statement.

UK7 Cat TL Clause 6.1.7, Page 7 lines 34 – 35 S-community

Rationale

The term is defined here, but the only reference to it in Clause 7 is effectively to repeat the definition. Thus "Rule 1[1]" applies, and the definition is redundant.

Proposal

Delete the definition (Clause 6.1.7), and the reference to the "s-community" in Clause 7 (see UK11)

6.1.8 C-object: A composite enterprise object that represents a community. Components of a c-object are objects of the community represented.

FIN22 E6.1.8 Community object

Rationale

Term under debate. The suggested term is one of the best alternative in the first rounds of discussion. Then it was not appropriate as there was a dangerously close counterpart with it, which now has totally disappeared.

Repetition that causes confusion.

Proposal

Rename with "community object".

[1] By tradition in the Working Group, "Rule 1" is taken to mean that no term should be defined in Clause 6 unless its usage is explained in Clause 7 or later.

Remove the latter sentence.

USA 10 TH Page 7, 6.1.8, lines 32-33 C-object

Rationale

The consensus that was reached in Paris on composition of communities does not require this concept.

Proposal

Remove: **6.1.8 C-object:** A composite enterprise object that represents a community. Components of a c-object are objects of the community represented.

6.1.9 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.

NOTES

1 – Examples of parties include enterprise objects representing natural persons, legal entities, governments and their parts, and other associations or groups of natural persons.

2 – Parties are responsible for their actions and the actions of their agents.

6.1.10 Machine: An enterprise object modelling an automated system.

NOTE – Examples of machines include computers and the software they support, systems of computers, other electronic devices and mechanical devices.

FIN23 TL 6.1.10 Machine

Rationale

All examples given are already concepts that can readily be used in any ODP system description as entities or can be described as ODP systems.

Proposal

Delete 6.1.10.

ES 5 TL Page 7, 6.1.10, line 44 Machine

Rationale

It is not clear here whether machines have responsibilities for their actions, or the actions of their agents. It is only later in Section 7.10.1 where an indirect reference is made to this.

Proposal

Include a second note after the definition of this concept:

NOTE – Machines do not have responsibilities for their actions, or the actions of their agents. This responsibility falls on the party or parties that delegated the actions to the machine.

TEMPORARY NOTE - The working group invites National Body comment on the need for this concept.

JP8 TL Page 7, 6.1.10, line 44 Machine

Rationale

The rationale to have this concept is not clear.

Proposal

Remove the concept.

Remove 6.1.10.

UK8 Cat TH Clause 6.1.10, page 7, line 44 Machine

Rationale:

The concept of "machine" appears to be aimed at distinguishing enterprise objects that represent automated systems from enterprise objects that could be "parties", in order to address the issue of "machine responsibility" e.g. a party can engage in "acts" but a machine cannot, except as an agent of a party.

However, the concept is an unnecessary complication of the model structure. The term "party" is appropriate for identifying enterprise objects that can have responsibility that has legal implications - though it is necessary for a specification that makes use of the concept to establish the legal context that applies. There is no need for a further term (i.e. "machine") to distinguish enterprise objects that that are not parties. From a review of the uses of the term "machine" (in 7.10, Force rules, page 22, lines 7, 8, 18, 20, 26, 27 and 36) there do not appear to be any cases where the term cannot be replaced by either by "ODP system" or simply "enterprise object".

Proposal:

a) Delete clause 6.1.10.

b) Make the corresponding changes to clause 7.10, Force rules. (See UK26)

USA 11 TH Page 7, 6.1.10, line 44 Machine

Rationale

Responding to the invitation to comment on the need for this concept: It is difficult and awkward to model machine responsibility without a concept with this meaning. The Part 2 concept, system, is not suited to this meaning, since 1) 'system' may be used both for entities in the universe of discourse and for elements of the model and 2) it is desired to use 'system' for systems that are not automated.

6.1.11 Owner (of an <X>): The party or one of several parties having the right to control the use and disposal of the <x>.

AFNOR 8 TH Page 8, Clause 6.1.11

Rationale

We think that the use of the notation "X" is not the same as in the part 2, where it means object as well as interface or action. We argue that here an X cannot be an action or an interface. So we must define what an X can be.

Proposal

6.1.11 Owner (of an <X>): The party or one of several parties having the right to control the use and disposal of the <x>. In RM-ODP, owner is needed for, at least, objects.

JP9 TL Page 8, 6.1.11, lines 4-5Owner (of an <X>)

Rationale

Simpler definition is preferred.

Proposal

Remove unnecessary part from the text.

Replace the current text with the following.

6.1.11 Owner (of an <X>): The party having the right to control the use and disposal of the <x>.

NOTES

1 – Commonly, this is a party paying for the specification, construction, instantiation, or current operation of the <x>. This party will typically grant authorisation to use the <x> to other parties or their agents.

2 – Ownership is restricted to parties to enable assignment of responsibility for actions.

USA 12 TL Page 8, 6.1.11, lines 4-9 Owner

Rationale

To be an owner is to fulfil a role.

Proposal

Move **Owner** to **6.2 Role Concepts**

6.2.n Owner (of an <X>): The role (of a party or one of several parties) having the right to control the use and disposal of that <x>.

NOTES

1 – Commonly, this role is fulfilled by the party (or parties) paying for the specification, construction, instantiation, or current operation of the <x>. This party will typically grant authorisation to use the <x> to other parties or their agents.

2 – Ownership is restricted to parties to enable assignment of responsibility for actions.

USA 13 TL Page 8, 6.1.11, line 7 Owner

Rationale

Payment may be a somewhat narrow concept, 'consideration' (from the contract law, see OED) may be a better term.

Proposal

Change:

~~paying~~ providing consideration for the specification, ...

Editor's Note - Part 2 uses '<X>' in both subclause headings and in body text. Part 3 (mostly) uses '<X>' (upper case) in subclause headings and <x> (lower case) in body text. This draft uses <x> in body text. The editor invites National Body comment on this question of style.

ES 6 E Page 8, Editor's Note, lines 10-13. <x> or <X> in body text

Rationale

Both <x> and <X> refer to the same thing. Many technical people are used to work in "case sensitive" environments, and therefore get easily confused trying to look for a (non-existent) difference between <x> and <X>. Furthermore, the angle brackets <> serve as a "escape" mechanism here, and therefore we do not see any problem for using capitals in body text.

Proposal

Use <X>, as in part 2.

FIN17 TH 6.1.x, 9.3 Scoping statement

Rationale

See clause 8 and capture that rationale and semantics into a definition.

Proposal

Add definition

A scoping statement specifies the preconditions for the use of the community. The scoping statement expresses either informally or formally what the community is intended to do and what properties the environment must have for the community to operate.

Remove 9.3.

6.2 Role concepts

FIN24 E6.2 Title

Rationale

The contents is mostly about objects instead of roles.

Proposal

Retitle as Role and enterprise object related concepts

6.2.1 Actor (with respect to an action): An enterprise object that participates in the action.

NOTE - It may be of interest to specify which of the actors involved initiates the action.

USA 14 TH Page 8, 6.2.1, line 16 Actor

Rationale

Several actors may collectively initiate an action.

Proposal

Change:

...which ~~of the~~ actors ~~involved~~ initiates the action

6.2.2 Artefact (with respect to an action): An enterprise object that is referenced in the action.

NOTE – An enterprise object that is an artefact in one action can be an actor in another action.

6.2.3 Resource: An artefact which is essential to some behaviour and which requires allocation or may become unavailable because it is in use or used up.

NOTE – A consumable resource may become unavailable after some amount of use.

ES 7 TL Page 8, 6.2.3, line 21 Note on Resource

Rationale

Resources may become unavailable after some amount of time (e.g., an expiry date), and not only by consumption.

Proposal

Change the Note to:

NOTE – A consumable resource may become unavailable after some amount of use, of after some amount of time (in case a duration or expiry date has been specified for the resource).

USA 15 TL Page 8, 6.2.3, lines 19-21 Resource

Rationale

A resource will often participate in the behaviour in which it fulfils the role of resource. In those cases it is not an artifact. (See page 18, lines 3-6.)

Proposal

Change:

6.2.3 Resource: An ~~artefact~~ object that is essential to some behaviour and which requires allocation or may become unavailable because it is in use or used up.

NOTE – A consumable resource may become unavailable after some amount of use.

USA 16 TH Page 8, 6.2.3, lines 19-21 Resource

Rationale

The definition appears to be ambiguous. Specifically, it is unclear whether because it is in use or used up" applies only to "may become unavailable". Also, the term "consumable" is introduced implicitly only in a note. Finally, in accordance with the definition of behaviour in 2-8.6, "essential to some behaviour" ought to be rephrased.

Lines 41-42 on page 17 should be appropriately rephrased.

Proposal

Change:

6.2.3 Resource An ~~artefact~~ object which is essential for some behaviour to occur and which requires allocation ~~or may become unavailable because it is in use or used up~~.

As a result of allocation, a resource may constrain the occurrence of other behaviours. A consumable resource may become unavailable after some amount of use.

6.2.4 Actor role (with respect to a community): A role in that community in which the enterprise object filling the role is involved in at least one action of the role as an actor.

6.2.5 Artefact role (with respect to a community): A role in that community in which the enterprise object filling the role is involved in all actions of the role only as an artefact.

6.2.6 Resource role (with respect to a community): A role in that community in which the enterprise object filling the role is involved in any actions of the role only as a resource.

TEMPORARY NOTE – The working group invites National Body comment on the need for these three role concepts.

JP10 TL Page 8, 6.2.4-6, lines 22-28 Actor/Artefact/Resource role

Rationale

With current definitions, if an enterprise object participating in an action works first as an actor, then as an artefact, then as resource, it would be called as object playing actor or resource role with respect to the community. How could this be useful in Enterprise viewpoint specification? We do not see the value of having current definitions.

Proposal

Remove those definitions or modify definitions.

Remove those definitions. Also remove line 8-16, page 18.

UK9 Cat TH 6.2, p6 lines 22-28 Actor, Artefact and Resource Roles

Rationale:

The UK, after some consideration, does not see the need for these concepts. In particular, the restriction, that an artefact role should only exist when the object concerned is only involved in actions as an artefact, seems artificial, unnecessary and is inconsistent with the definitions of actor role and resource role (which exist whenever an object is involved in an action as an actor or resource respectively).

Proposal:

Delete clauses 6.2.4, 6.2.5 and 6.2.6, and the temporary note that follows them.

USA 17 TL Page 8, 6.2.6, lines 26-27 Resource role

Rationale

'any actions … only as' is unclear. 'Any' might mean any one or might mean all.

Proposal

Change:

6.2.6 Resource role (with respect to a community): A role in that community in which the enterprise object filling the role is involved in all actions of the role only as a resource.

USA 18 TL Page 8, 6.2.1-6.2.6, lines 15-28 Actor, &c. role

Rationale

Responding to the invitation to comment for the need for these three role concepts: This was an earlier ballot comment:

"The style of specification we have used uses actor roles with respect of the community and itemizes artefact roles with respect to each individual action. We suggest that only the terms actor and artefact were defined, and additional sentences are used to explain that there is a need to indicate whether the use is with respect to an action or with respect to a community. Each specification should be consistent in the chosen style, but the choice rests with text organisation decisions within individual tools that help in composing enterprise specifications."

This was rejected, as lacking replacement text.

We make these additional observations:

1) As we understand the ITU rules, use of the parenthetical form, '(with respect to a community)' restricts the ways in which the term can be used. We do not see a reason to so restrict the use of these terms.

2) The definitions prohibit many reasonable usages. For example, if a role includes on action in which the object fulfilling the role is an actor, it is not permitted to say, for example, of a role which provides an important resource to a community, that "this is a resource role for this community."

As the working group knows, in an enterprise specification the term '<x> object', where <x> is a role, is interpreted as meaning an enterprise object fulfilling an <x> role. [3-5.2]

We believe this proposal meets the objective of including the concepts, actor, artefact, and resource in the language, answers the concerns of the earlier comments shown above and deals with our observations above.

We believe it is closer to the original intent of the concepts to consider actor and artefact to be roles.

Incidentally, this conforms to the UML actor, which is a role, rather than an object, a small bonus at no extra cost.

Proposal

Replace 6.2.1 thru 6.2.6 with:

6.2.1 Actor: A role that includes at least one action in which the object fulfilling the role participates.

NOTE – It may be of interest to specify which of the actors involved in such an action initiates that action.

6.2.2 Artefact: A role in which the object fulfilling the role is, in every action of the role, referenced, but does not participate.

NOTE – An enterprise object that is an artefact in one role can be an actor in another role.

6.2.3 Resource: A role that is essential for some behaviour to occur and which requires allocation or may become unavailable because it is in use or used up.

As a result of allocation, a resource may constrain the occurrence of other behaviours. A consumable resource may become unavailable after some amount of use.

6.2.7 Interface role (with respect to a community): A role of a community that identifies actions that can be performed by a c-object representing that community.

Editor's Note - Should this be interactions?

FIN26 TH 6.2.7 Answer to editor question

Proposal

Should use interactions instead of actions.

ES 8 TL Page 8, 6.2.7, lines 29-30 Interface Role (wrt a community)

Rationale

The "actions" mentioned in that definition should be refined to "interactions" and the "constraints" governing those interactions. This would make this definition more consistent with the definition of "interface" in II-8.4.

Proposal

Change definition 6.2.7 to:

6.2.7 Interface role (with respect to a community): A role of a community that identifies the actions, in terms of interactions and the constraints on when they may occur, that can be performed by a c-object representing that community

USA 20 TH Page 8, 6.2.7, lines 29-31 Interface roles

Rationale

The consensus that was reached in Paris on composition of communities has not been carried through in the text.

Proposal

We propose that an e-mail meeting be started to work out an explanation of interface roles and interactions between communities that corresponds to the consensus reached in Paris. At the same time, language can be worked out that more clearly explains the approach of this text.

USA 21 TH Page 8, 6.2.7, lines 29-31 Interface roles

Rationale

The actions referred to in the text are performed collectively by the c-object and some other objects. It is too restrictive to request a performer for each action (collective behaviour does not have this restriction). See also Editor's note.

Proposal

Change: ...actions that can be performed by a c-object representing that community

To:

...interactions in which a c-object representing that community participates.

6.2.8 Agent: An enterprise object that has been delegated (authority, responsibility, a function, etc.) by and acts for another enterprise object (in exercising the authority, carrying out the responsibility, performing the function, etc.).

NOTES –

1 – An agent may be a party or may be the ODP system or one of its components. Another system in the environment of the ODP system may also be an agent.

2 – The delegation may have been direct, by a party, or indirect, by an agent of the party having authorisation from the party to so delegate.

TEMPORARY NOTE – This term is intended to have a meaning that follows one standard meaning of 'agent,' that in the pair agent/principal. This meaning may be different from a currently popular use of 'agent' in computer software circles, which may be closer to that in the pair agent/patient. In Madrid, the working group decided not to change this term.

USA 22 TL Page 8, 6.2.8, lines 32-43 Agent

Rationale

The definition may be somewhat different from the one implied by 3-5.2 Note: "An important special case of acquisition is where the permitted action is performative, i.e., when an object in a subordinate role is enabled to issue further permissions or obligations on behalf of an object fulfilling a superior role. This leads to the notion of agency or delegation." Is this the case? Is there a need to refine or change that Note?

Proposal

?

6.2.9 Principal: A party that has delegated (authority, a function, etc.) to another.

6.2.10 Contracting party (with respect to a contract): A party that agrees to that contract.

FIN25 TH 6.2.x Assignment rule

Rationale

The population process of communities is not described properly, partly because the basic concepts are not yet defined.

Proposal

Add

Assignment rule: denotes the criteria on which an enterprise object is selected to perform a role. The criteria defines what the object to fulfil a role must be capable of doing, not restricted from doing by earlier commitments, and what relationships to other objects are required or denied.

6.3 Policy concepts

6.3.1 Policy: A set of rules related to a particular purpose. A rule can be expressed as an obligation, an authorisation, a permission or a prohibition.

NOTES:

1 – Not every policy is a constraint. Some policies represent an empowerment.

2 – This definition refines 2-11.2.7.

FIN27 TL 6.3.1 Authorization

Rationale

Authorization is one of the force concepts, and belongs to a different kettle than permission, prohibition and obligation.

Proposal

Remove authorisation from line 4.

Remove note 1.

6.3.2 Authorisation: A prescription that a particular behaviour must not be prevented.

NOTE – Unlike a permission, an authorisation is an empowerment

FIN28 TL 6.3.2 Authorization

Rationale

In distributed environment, there are no mechanisms that can be used to guarantee this in a general case. Although the concept is useful in special cases, introducing it here gives faulty expectations. When specifiers need to define their own authorization concepts they also need to consider whether that is actually possible within the circumstances they have chosen, or not.

Proposal

Remove 6.3.2.

6.3.3 Violation: An action contrary to a rule.

NOTE – A rule or policy may provide behaviour to occur upon violation of that rule or policy.

TEMPORARY NOTE – The working group invites National Body comment on structuring rules using this concept.

FIN30 E6.3.3 Violation

Rationale

There are violations against explicitly stated policies, i.e., failures (violations against a contract). These should be considered as errorousness behaviour implementation or specification. There are also violations caused by inconsistent assumptions between communicating parties about the permissions, obligations and prohibitions. These may arise for example in federation situations where there is no full control to the interacting objects environment or in other situations where an action is not considered to be essential enough to be specified with policies in detail for all possible participants of an interaction. Only some of these situations can be considered as design or implementation errors. Mechanisms are needed for catching violations and for using appropriate recovery or sanction mechanisms.

Proposal

Rephrase

An event causes a violation situation to arise when their expectations on permissions, prohibitions or obligations on an action do not match.

Notes –

Violations occur while an enterprise object tries to follow requirements associated to its role within either a single community or within multiple communities.

Failure is a violation against a consistent contract.

Violations can also occur because of inconsistencies between the contracts involved.

There can be policies or rules expressed for a community or for a federation defining recovery or sanction mechanisms for failures and other violations.

USA 23 TH Page 9, 6.3.3, line 11 Violation

Rationale

A rule may provide behaviour to occur upon violation of that, or some other, rule or policy. The existing text is too restrictive.

Question: Is it possible to prescribe rules or policies to be used upon violation of specific rules or policies? If so then the Note referred to above ought to be rephrased appropriately.

Proposal

Change:

A rule or policy may provide behaviour to occur upon violation of ~~that~~ some rule or policy.

FIN29 TH 6.3.x Policy framework

Rationale

The concepts of objective, cobehaviour, policy and nesting policy frameworks are poorly understood, mostly because an essential structuring concept is missing.

Proposal

Add

Policy framework: A structuring rule in a community specification that specifies for which purposes policies are set in these communities.

Notes -

1 - When communities are nested, the enterprise objects must conform to all policies in all communities it participates. As the participated communities may have different policy frameworks, the effective set of policies becomes larger.

2 - When a community plays a role in another community, the policies may not be contradictory, but the policy frameworks may differ.

3 - Establishing a federation means creation of a new community, thus, a new policy framework is created that is consistent with the policy frameworks of the establishing parties. For example, the members of the federated community must conform to both the communities within their organizations and the federated community spanning across the organizatonal boundary. The federated community policy framework may leave some aspects of common behaviour without policies. Therefore, enterprise objets may run into situations where their policies contradict.

6.4 Force concepts

These concepts may be used to model changes in that part of the universe of discourse modeled by the environment of the ODP system, including such changes when caused by changes in the ODP system itself.

6.4.1 Act: A behaviour of parties or agents initiated by a purposeful selection.

NOTES:

1 - An act is therefore intentional, chosen to suit an objective.

2 - The enterprise object or objects participating in an act may be parties or machines acting as agents.

FIN31 TH 6.4.1 Purposeful selection
Rationale
Remove references to purposeful selection.
Proposal
Rewrite as:
A behaviour of parties or agents related to community objective.
Remove both notes.

6.4.2 Commitment: An act resulting in an obligation by one or more of the participants in the act to comply with a rule or perform a contract.

NOTE – The enterprise object(s) participating in an act of commitment may be parties or agents acting on behalf of a party or parties. In the case of an act of commitment by an agent, the principal becomes obligated.

6.4.3 Declaration: An act that establishes the truth of a sentence referring to the environment of the object making the declaration.

NOTE – The essence of a declaration is that, by virtue of the act of declaration itself, it causes a state of affairs to come into existence outside the object making the declaration.

FIN32 TH 6.4.3 declaration
Rationale
The concept of truth is subjective. In a distributed environment, there are no mechanism to force a given state of affairs to take effect on a domain where the statement maker has no authority.
Proposal
Rewrite as:
An act by which a party publishes to other parties a statement about the state of affairs the party is committed to support.
Note – State of affairs may be related to the ODP system state, environment state, or future behaviour of the system or environment. The statement may be about a community or an enterprise object in the system that is controllable by the party or outside the control of this party.

USA 19 TH Page 9, 6.4.3, lines 24-25 Declaration

Rationale

Conform to the principle that truth applies to propositions, rather than the sentences which express them.

Proposal

Change:

6.4.3 Declaration: An act that establishes the truth of a ~~sentence~~ proposition referring to the environment of the object making the declaration.

6.4.4 Delegation: The action of assigning authority, responsibility or a function to another object.

JP11 TL Page 9, 6.4.4, line 28 Delegation

Rationale

Delegation can also be behaviour of parties or agents initiated by purposeful selection.

Proposal

Remove inappropriate word from the text.

Replace current text with the following.

6.4.4 Delegation: An act that assigns authority, responsibility or a function to another object.

NOTES:

1 - Delegation may be from a party to a machine.

ES 9 TL Page 9, 6.4.4, line 30 Note 1 on Delegation

Rationale

Delegation may also be from a machine to a party, as it happens when a program instructs (prompts) a user to do something. In this case, responsibilities should be made clear at this point (see our comment ES5).

Proposal

Change Note 1 to:

1 – Delegation may be from a party to a machine, or from a machine to a party, as it happens when a software program instructs a user to do something. In case of delegation from a machine to a party, the responsibility of the final action falls on the party that initially delegated to the machine the action that caused the final delegation.

2 - Delegation may assign responsibility for making purposeful selections. In this case the action of delegation provides the objective and may provide the selection criteria.

FIN33 E6.4.4 delegation

Proposal

Replace note 1-2 with

Note – Delegation may be from a party to a system and the system may also be delegated an authority to make further delegations.

USA 24 TL Page 9, 6.4.4, line 33 Delegation

Rationale

This is responsive to a comment on the previous ballot which was marked as noted, but resulted in no change to the draft.

Proposal

Add:

NOTES:

3 – A delegation, once made, may be withdrawn.

Editor's Note - We are leaving unspoken the other possibility raised by RM-ODP: transfer, as opposed to delegation. [3-5.2]

ES 10 EPage 9, 6.4.4, lines 33-34 Editor's Note on Delegation

Rationale

The editor's note on delegation suggests that "transfer" is another mechanism used in ODP (III-5.2) which may be missing in this part.

Proposal

The differences between "transfer" and "delegation" need to be clarified here, and a new concept "transfer" should be defined if the differences are significant enough. If not, a note should mention that "transfer" may be used instead of delegation as a mechanism for assigning authority, responsibility, functions, permissions, obligations, or prohibitions.

6.4.5 Evaluation: An act that assigns a value to something.

NOTE – For example, the act by which an ODP system assigns a relative status to some thing, according to estimation by the system of its worth, usefulness, or importance.

FIN34 E6.4.5 Evaluation

Rationale

Ambiguity.

Proposal

Rewrite as:

An act that assesses the value of something.

Note – value can be considered in terms of usefulness, importance, preference, acceptabilty etc; the evaluated target can be a system state, a potential behaviour, etc.

6.4.6 Prescription: An act that establishes a rule.

7 Structuring Rules

USA 3 G TH Composition of communities

Rationale

The consensus that was reached in Paris on composition of communities has not been carried through in the text.

Proposal

Follow the approach explained by Reenskaug in Paris.

If the working group does not accept this proposal, then the text needs to get more clear and specific about the alternative approach to composition of communities. In particular, the standard needs to clearly specify what it means for an object to participate in an action of a role of a community when that object is not a member of that community.

If there are going to be two approaches to composition of communities, this needs to be made explicit and each needs be described separately.

7.1 Overall structure of an enterprise specification

An enterprise specification for an ODP system is a model of that system and relevant parts of its environment. The enterprise specification focuses on the scope and purpose of that ODP system and the policies that apply to it in the context of its environment.

ES 11 EPage 10, 7.1, line 3 "specification" vs. "model"

Rationale

The definition of an enterprise specification in page 10 says: "An enterprise *specification* for an ODP system is a *model* of that system and relevant parts of its environment". However, an specification and a model are two different things, as discussed by Guy Genilloud, Janis Putman, and Sandy Tyndale-Biscoe in their e-mails to the enterprise@dstc.edu.au list.

Proposal

In order to avoid any possible confusion, we suggest eliminating the word "model" from that definition, changing the first paragraph in section 7.1 to:

An enterprise specification for an ODP system is a detailed description of the system and relevant parts of its environment, focusing on the scope and purpose of that ODP system, and the policies that apply to it in the context of its environment.

UK10 Cat E Clause 7.1, Page 10, line 3 Overall structure of an enterprise specification

Rationale:

See UK2 on the use of "enterprise specification" and "model".

Proposal:

Make the following change to Page 10, line 3 by:

An enterprise specification for an ODP system expresses~~is~~ a model of that system and relevant parts of its environment.

USA 25 TH Page 10, 7.1, lines 3-43 Overall structure

Rationale

This clause mixes explanatory material with structuring rules.

Proposal

Move text that is not a structuring rule to the explanatory annex.

FIN37 E7.1, between lines 5 and 6 Nature of ODP systems

Rationale

The nature of ODP systems is not clear.

Proposal

Add:

The ODP system under specification may span multiple organizations.

A fundamental structuring concept for enterprise specifications is that of community. A community is a configuration of enterprise objects modelling a collection of entities (e.g. human beings, information processing systems, resources of various kinds and collections of these) that are subject to some implicit or explicit contract governing their collective behaviour.

USA 26 EPage 10, 7.1, lines 6-9 Overall structure

Rationale

The draft text omits the essential concept of community, objective.

Proposal

Change:

A community is a configuration of enterprise objects modelling 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, which entities are subject to some implicit or explicit contract governing their collective behaviour.

USA 27 TL Page 10, 7.1, lines 6-14 Overall structure of an enterprise specification

Rationale

On the one hand, a community is described as a configuration of enterprise objects without any reference to roles. On the other hand, immediately afterwards the ODP system is described in terms of roles in communities, but it is not clear here how roles participate in a community specification. Thus, the description here ought to be clarified.

Proposal

?

USA 28 TH Page 10, 7.1, lines 6-9 Overall structure

Rationale

The draft text unnecessarily combines a discussion of the universe of discourse with structuring rules.

The text unnecessarily gives examples and thereby may be read to restrict the specifier's use of community.

The contract (or contracts) governing the collective behaviour has to be specified explicitly; this follows from the definition of a contract in 2-11.2.1 and from the need to be explicit about the constraints imposed by the contracts on the community participants. If the contract remains implicit then reasoning about the system in the context of its communities becomes impossible ("miracles happen"). Those fragments of the contract(s) that are considered irrelevant for the enterprise specification may be suppressed by appropriate usage of abstraction (2-6.3).

Note. The same considerations apply to page 11, line 20.

Proposal

Change:

A community is a configuration of enterprise objects ~~modelling 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. Those objects are subject to ~~some implicit or explicit~~ a contract governing their collective behaviour. That contract specifies how the objective can be met.

FIN38 E7.1, between lines 9-10 Levels of abstraction

Proposal

Add:

Note – The set of community specifications selected may be induced by using various levels of abstraction, functional decomposition of the system and ownership of the system or parts of the system.

The ODP system may play a role in more than one community. Thus, the enterprise specification describes, within the areas of interest of the specification users:

FIN35 E7, page 10, line 17 contradictory statements

Rationale

C-community does not exist; note on lines 19-20 contradictory with lines 17-18.

Proposal

Line 10: The ODP system may participate more than one communities.

USA 29

No comment.

USA 30 EPage 10, 7.1, line 10 Overall structure

Rationale

Usage.

Proposal

Change:

The ODP system may play a role in more than one community. ~~Thus,~~ The enterprise specification describes …

USA 31 TL Page 10, line 10 7.1 Overall structure of an enterprise specification

Rationale

The text elsewhere consistently uses “roles fulfilled by objects” rather than “roles played by objects”.

Proposal

Change:

The ODP system may ~~play~~ fulfill a role...

-- roles fulfilled by the ODP system;

AFNOR 9 TH page 10, Clause 7.1, line 12

Rationale

An enterprise specification is about a system and its environment.

Proposal

Change to :

-- roles fulfilled by the ODP system and its environment;

-- activities undertaken by the ODP system within processes in which it participates;

-- policy statements about the system, including those relating to environment contracts.

USA 32 TH Page 10, 7.1, line 13 Overall structure

Rationale

The draft language introduces a new concept, ‘activity within a process,’ and contradicts the text at 6.1.5 Note 2.

Proposal

Change:

-- ~~activities undertaken by the ODP system~~ steps within processes in which the ODP system participates;

or

-- ~~activities undertaken~~ by the ODP system within steps of processes in which ~~it~~ the ODP system participates;

USA 33 TH Page 10, 7.1, line 14 Overall structure

Rationale

The draft language introduces a new useage, 'policies about.'

Proposal

Change:

-- ~~policy statements~~ policies ~~about~~ for the system, including those relating to environment contracts.

An enterprise specification of an ODP system includes at least a description of the community in which that system may be represented as a single enterprise object interacting with its environment. This is referred to as the s-community. Whether the specification actually includes that level of abstraction is left for the specifier to decide.

FIN35b E7, page 10, line 17 contradictory statements

Rationale

C-community does not exist; note on lines 19-20 contradictory with lines 17-18.

Proposal

…

Line 17: Delete sentence.

Line 17: replace c-community with "root community specification".

Line 17: roles, processes -> roles or processes (too strong requirement otherwise)

Note by ST-B: I cannot locate the text referred to here; probably "c-community" should be "s-community", which is mentioned in line 17, but there is no reference to roles or processes here.

AFNOR 10 E page 10, Clause 7.1, line 15

Rationale

See AFNOR 1

Proposal

Change line 15 to:

An enterprise specification of an ODP system includes at least the community in which that system may be represented as a single enterprise object interacting with its environment.

AFNOR 1 G A community is a specification

Rationale

As an object is a model of an entity, a community is a model of a collection of entities. Thus, a community is a specification.

Proposal

Everywhere in the standard, change “community specification” or “description of a community” by “community”. For example, page 10, clause 7.1, line 21, change to “the enterprise specification can include any other communities “

UK11 Cat TL Clause 7.1, Page 10 lines 16 - 17 Overall structure of an enterprise specification

Rationale

See UK7

Proposal

Delete the sentence reading “This is referred to as the s-community”.

USA 34 EPage 10, 7.1, lines 15-18 Overall structure

Rationale

Style

Proposal

Change:

An enterprise specification of an ODP system includes ~~at least~~ a description of the community in which that system may be represented as a single enterprise object interacting with its environment. This is referred to as the s-community. Whether the specification ~~actually~~ includes that level of abstraction is left for the specifier to decide.

USA 35 TH Page 10, 7.1, lines 15-18 Overall structure

Rationale

The draft text contradicts itself.

Proposal

Change:

An enterprise specification of an ODP system includes ~~at least~~ a description of the community in which that system may be represented as a single enterprise object interacting with its environment. This is referred to as the s-community.~~ Whether the specification actually includes that level of abstraction is left for the specifier to decide.~~

Or:

An enterprise specification of an ODP system includes at least a description of the community in which that system may be represented as a single enterprise object interacting with its environment. ~~This is referred to as the s-community. Whether the specification actually includes that level of abstraction is left for the specifier to decide.~~ The specifier may choose to decompose that ODP system in the specification of that community.

USA 36 TL Page 10, lines 15-18 7.1 Overall structure of an enterprise specification

Rationale

There may be more than one community in which the ODP system may be represented as a single enterprise object. The term "s-community" appears not to be used anywhere else and thus may not be explicitly needed.

Proposal

Consider deletion of 6.1.7.

NOTE – This minimal enterprise specification details the objective and scope of the ODP system and is necessary for completeness of the enterprise specification.

Where necessary for clarity or completeness, the enterprise specification can include descriptions of any other communities of which the ODP system or its components are members, and other communities of which enterprise objects in the environment of the ODP system are members.

AFNOR 11 E page 10, Clause 7.1, line 21

Rationale

See AFNOR 1

Proposal

Change line 21 to :

Where necessary for clarity or completeness, the enterprise can include any other communities of which the ODP system or […]

NOTE – In order to understand the ODP system behaviour, it may be necessary to describe communities at both more abstract and more detailed levels than the minimal enterprise specification. These communities may have objectives that are inconsistent with the objective of the ODP system itself.

The enterprise specification can also be structured in terms of a number of communities interacting with each other. In such a case the communities concerned, viewed as composite objects (c-objects), themselves form a community that may be described explicitly or may be left implicit and the ODP system must be a member of at least one of those communities

NOTE – Such structuring may represent, for example, co-operation of domains in a federation.

The scope of the system is defined in terms of its intended behaviour and in enterprise language this is expressed in terms of roles, processes, policies and their relationships.

USA 37 TH Page 10, 7.1, lines 32-33 Overall structure

Rationale

Community and contract are two key concepts of the enterprise language, given by Part 3.

Proposal

Change:

The scope of the system is defined in terms of its intended behaviour and in enterprise language this is expressed in terms of communities, roles, contracts, processes, policies and their relationships.

USA 38 TL Page 10, 7.1, lines 32-33 Overall structure

Rationale

Scope is defined as expected behaviour.

Proposal

Change:

The scope of the system is defined in terms of its ~~intended~~ expected behaviour and in the enterprise language this is expressed in terms of roles, processes, policies and their relationships.

A complete ODP system specification indicates rules for internal consistency in terms of relationships between various viewpoint specifications. Furthermore, a complete enterprise specification contains conformance rules that define the required behaviour of the described ODP system, thus separating out the parts of the specification included for the purposes of analysis, understandability and communicability of the specification.

USA 39 TH Page 10, 7.1, lines 36-37 Overall structure of an enterprise specification

Rationale

Analysis is "breaking down a whole into its components and their mutual relations" (Bunge). Thus, it is not clear how to separate parts of a specification included for the purposes of analysis. Similarly, it is difficult to separate the parts of the specification included for the purposes of understandability and communicability since such separation may imply that some (other) parts of the specification are included for the purposes of not being understood or not being communicable.

Proposal

Change:

A complete ODP system specification indicates rules for internal consistency in terms of relationships between various viewpoint specifications. Furthermore, a complete enterprise specification contains conformance rules that define the required behaviour of the described ODP system~~, thus separating out the parts of the specification included for the purposes of analysis, understandability and communicability of the specification~~.

This clause defines how the concepts identified in clause 3 or defined in clause 6 of this Recommendation | International Standard are used in an enterprise specification.

NOTE – This Recommendation | International Standard makes no prescriptions about either the most detailed or the most abstract levels of any enterprise specification, nor does it make any recommendations about the relative merits of modelling from 'top-down' or 'bottom-up'. The approach taken is a modelling choice based on the ODP system being specified and the purpose of the modelling.

FIN36 E7.1 Repeating text

Rationale

Repeated in 7.2

Proposal

Delete final note (lines 40-43) from 7.1

USA 40 TL Page 10, lines 40-43 7.1 Overall structure of an enterprise specification

Rationale

This text is presented, perhaps in a better way, in Clause 7.2

Proposal

Remove:

NOTE – This Recommendation | International Standard makes no prescriptions about either the most detailed or the most abstract levels of any enterprise specification, nor does it make any recommendations about the relative merits of modelling from 'top-down' or 'bottom-up'. The approach taken is a modelling choice based on the ODP system being specified and the purpose of the modelling.

7.2 Contents of an enterprise specification

An enterprise specification is composed of specifications of the elements explained in clause 7.1 (communities, objects, roles, behaviour, …).

AFNOR 12 E page 11, clause 7.2, line 2

Rationale

See AFNOR 1

Proposal

Change line 1 to:

An enterprise specification is composed of the elements explained in clause 7.1 (communities, object, roles, behaviour, …)

UK12 Cat E Clause 7.2, P11, line 3 Contents of an enterprise specification

Rationale

The use of the ellipsis in the list in parentheses leaves the reader unsure which are the concepts being described.

Proposal

Replace the text in parentheses with:

"(communities, enterprise objects, contracts, roles, activities, behaviour, the ODP system, policies, and interactions)."

USA 41 TH Page 11, 7.2, lines 2-4 Contents of an enterprise specification

Rationale

The elements referred to in these lines were not explained in Clause 7.1; at best, they were referred to. Not all concepts referred to in 7.1 are included as elements (e.g., scope is not included, and it does not

follow from 7.2 that scope is included in the contents of the enterprise specification). It may be reasonable and useful to provide a set of the possible elements (i.e., to complete the enumeration). Moreover, each of these elements does not exist in isolation, and therefore relationships between these elements ought to be explicitly referred to; it is possible to include such relationships as elements since they also may have types and templates.

Note. It may be worthwhile to replace "elements" with "components" elsewhere, for example, in line 13 of page 12.

Proposal

Change: An enterprise specification is composed of specifications of the elements explained in clause 7.1 (communities, objects, roles, behaviour, ...).

Depending on the specifier's choice and desired level of detail, each of these elements may be specified by

To:

An enterprise specification is composed of specifications of communities, contracts, objects, roles, actions, behaviour, etc., and relationships between them. The characteristics of this composition include the purpose, scope and policies of the ODP system.

Depending on the specifier's choice, each of these components may be specified by

Depending on the specifier's choice and desired level of detail, each of these elements can be specified by

AFNOR 13 E page 11, clause 7.2, line 4

Rationale

See AFNOR 1

Proposal

Change line 4 to:

Depending on the specifier's choice and desired level of detail, the enterprise specification is composed of

-- the characteristics of the element, or

-- the type or types of the element, or

-- a template of the element.

The enterprise language makes no prescription about the specification process nor the level of abstraction to be used in an enterprise specification.

NOTES:

1 – No recommendations are made about the relative merits of modelling from top-down or bottom-up. Nor is there a recommended sequencing of the development of viewpoint specifications.

ES 12 E Page 11, 7.2, Note 1, lines 11-12 top-down/bottom-up modelling

Rationale

The content of note 1 has been already mentioned in a previous note (the one at the end of page 10, lines 40-43), and therefore seems to be redundant.

Proposal

Remove Note 1.

2 – It is a design choice whether a specification deals with a specific implementation by, for example, identifying individual enterprise objects, or deals with a more flexible architecture by identifying types and assignment rules for accepting enterprise objects.

Editor's Note - Does 'assignment rules for accepting enterprise objects' mean rules for assigning enterprise object to roles?

FIN39 TH 7.2 Assignment rules

Rationale

Answer to editors question: Yes.

ES 13 E Page 11, 7.2, Editor's Note to Note 2, lines 13-17

Rationale

We agree with the editor's note, and think that it is worth rewording the note 2 according to his suggestions.

Proposal

Change Note 2 (lines 13-15, page 11) to:

2 – It is a design choice whether a specification deals with a specific implementation by, for example, identifying individual enterprise objects, or deals with a more flexible architecture by identifying types and rules for assigning enterprise object to roles.

UK13a

a) Move Page 13, lines 1-2, lines 4-5 (Note 1) and lines 11-13 (Note 4) to 7.2 to follow the existing Notes on Page 11, lines 10-15 as additional Notes:

3 – A specification may be partitioned because of readability, reuse of specification fragments in other specifications or interoperability of enterprise objects.

4 – Role and community specifications, as well as type and template specifications, can be private to a specification and development environment, or they can be stored to a repository that can be shared by a wider audience of several development environments and groups.

7.3 Community rules

7.3.1 Specification of a community

AFNOR 14 E page 11, clause 7.3.1, line 19

Rationale

See AFNOR 1

Proposal

Change line 19 to:

7.3.1.Contents of a community

For a set of entities to be modelled as a community there must be some implicit or explicit agreement about the collection covering the reason for its existence, how it is organised and what it does, and what entities comprise its membership. This agreement is expressed as the contract for the community that

ES 14 EPage 11, 7.3.1, line 20 Specification of a community

Rationale

As mentioned in our comment ES11, we agree with Guy Genilloud's comment that specification and model correspond to two different concepts.

Proposal

Change the first paragraph:

For a set of entities to be modelled as a community there must be some implicit or explicit agreement about the collection covering the reason for its existence, how it is organised and what it does, and what entities comprise its membership. This agreement is expressed as the contract for the community

To:

For a set of entities to be specified as a community there must be some implicit or explicit agreement about the collection, covering the reason for its existence, how it is organised and what it does, and what entities comprise its membership. This agreement is expressed as the contract for the community

USA 42 TH Page 11, 7.3.1, lines 20-22 Specification of a community

Rationale

In its comments with the previous CD ballot, the USA gave this rationale: The only reason for a set of entities to be modelled as a community will be some reason of the person making the specification. That is to say: it is up to that person to decide which entities are in the universe of discourse, and which sets of those entities will be modelled as members of communities. Generally, there need be no explicit or implicit agreement between the entities.

In the interest of moving the meeting to other topics, the proposal was withdrawn at that meeting.

We make this proposal again, and add these comments:

It is true that the agreement might be between entities other than those modelled as objects in the community, e.g. designers or developers.

Still: USA position is: It is up to the person making the specification to chose which entities are in the universe of discourse, to choose which of those entities are modelled in the enterprise specification (that is, appear as enterprise objects in that specification), and to choose which sets of those entities are modelled as communities.

This standard has no business limiting these choices.

Proposal

Replace: For a set of entities to be modelled as a community there must be some implicit or explicit agreement about the set covering the reason for its existence, how it is organised, what it does, and what entities comprise its membership. This agreement is expressed as the contract for the community that:

With:

The objective of a community is expressed as a contract that specifies how the objective can be met. This contract:

USA 43 TH Page 11, 7.3.1, lines 20-22 Specification of a community

Rationale

This is an alternative to the proposal in the preceding comment.

It may be that the intent of this structuring rule is this: every community must have a contract which specifies how its objective can be met. If so, the rule should state that intent.

Proposal

If the preceding comment is not accepted: ***Replace:*** For a set of entities to be modelled as a community there must be some implicit or explicit agreement about the set covering the reason for its existence, how it is organised, what it does, and what entities comprise its membership. This agreement is expressed as the contract for the community that:

With:

The specification of a community states the reason for its existence, how it is organised, what it does, and what objects comprise it. The objective of the community is expressed as a contract that specifies how the objective can be met. This contract:

USA 44 TH Page 11, 7.3.1, lines 20-22 Specification of a community

Rationale

The concept, 'structure of a community' is not defined.

Proposal

Change:

-- governs the ~~structure~~ configuration, the behaviour and the policies of the community,

-- states the objective for which the community exists,

-- governs the structure, the behaviour and the policies of the community,

-- constrains the behaviour of the members of the community,

-- states the assignment of enterprise objects to roles.

FIN41 TH 7.3.1, page 11, line 26 Population process

Rationale

Hopefully just editorial.

Proposal

Replace "States the assignment of enterprise objects to role" with "States the assignment rules of enterprise objects to role".

The contract can be formed either by a defined process carried out by some or all of the enterprise objects at the time of community establishment, or in an earlier epoch (for example, as a design decision).

AFNOR 16 TH page 11, clause 7.3.1, line 27-28

Rationale

The contract cannot be defined by enterprise objects because they are only models. Moreover, the establishment of the contract is out of the scope of this standard.

Proposal

Remove these lines and the notes.

USA 45 TH Page 11, 7.3.1, lines 27-28 Specification of a community

Rationale

The specifier need not use a process in the sense defined in the language. The objects forming the contract need not be objects of the community.

Proposal

Change:

The contract can be formed either by a defined ~~process~~ behaviour carried out by ~~some or all of the~~ enterprise objects at the time of community establishment, or in an earlier epoch (for example, as a design decision).

NOTES

1 – The concept, contract, is defined in ITU-T X.902 | ISO/IEC 10746-2.

ES 15 EPage 11, 7.3.1, Note 1, line 30 Note 1

Rationale

We do not see the need for this special treatment of concept "contract", explicitly mentioning where it is defined. We propose to treat it as the rest of the concepts used in this recommendation that have been defined in other recommendations of ODP.

Proposal

Remove Note 1.

2 – There is no requirement that the parties forming the contract fulfil roles in the community. Indeed, the lifetime of a community can extend beyond the lifetime of the parties to the community contract.

ES 16 EPage 11, 7.3.1, Note 2, lines 31-36 Note 2 and editor's note

Rationale

Note 2 is difficult to understand as it is.

Proposal

Try to re-write Note 2 in order to clarify its intention and contents. Unfortunately we do cannot propose any alternative text because we do not fully understand the Note.

Editor's Note – Note 2 (no parties to contract need fulfill roles) might be read as contradicting the normative text (which suggests that at least some of the enterprise objects of the community ("the enterprise objects") form the contract, if it is formed at the time of community establishment.

FIN42 E7.3.1, page 11, lines 27-36 Establishing a community

Rationale

Text is ambiguous and difficult to interpret.

Proposal

Replace with

The contract is formed by a defined process carried out by a community that has the objective of forming a new contract. The members of this establishing community include parties, agents representing these parties, or members of a new community exploiting the contract-to-be. This establishing community appears in an earlier epoch than any community based on the new contract. This earlier epoch may represent a defined design process, defined negotiation process between agents, or commitment of agents to a new community.

Note - The first epoch involved is a design process.

The behaviour of the community expresses how it meets its objective. In the context of the community, the objectives of members of the community are constrained to conform to its objective.

USA 46 TL Page 11, 7.3.1, line 37 Specification of a community

Rationale

The behavior ***is*** how the community meets its objective.

Proposal

Change:

The behaviour of the community ~~expresses~~ is how it meets its objective.

or delete.

USA 47 TL Page 11, 7.3.1, lines 37-38 Specification of a community

Rationale

The concept, 'conform to an objective,' raises the question, what it means to conform to an objective. The alternative text proposed below express what may be intended. "A contract specifies [rules] for the objects involved." [2-11.2.1] "The objective [of a community] is expressed as a contract which specifies how the objective can be met." [3-5.1.1]

Proposal

Remove: In the context of the community, the objectives of members of the community are constrained to conform to its objective.

Or: ***Replace with***:

The objects of a community are constrained by the rules of the community contract.

The behaviour of the community is defined in terms of one or more of the following elements:

-- the roles of the community (including those roles which define how a community interacts with its environment),

-- the processes that take place in the community,

-- the way the roles and processes are combined and,

FIN43 E7.3.1, page 12, line 1 Use of EL

Rationale

Defines methodology.

Proposal

Combined -> related

USA 48 TH Page 12, 7.3.1, line 1 Specification of a community

Rationale

Unclear and not specific.

Proposal

Replace: -- the way the roles and processes are combined and,

With:

-- the assignment of roles to steps in processes and,

-- policies that apply to the roles and processes.

ES 17 EPage 12, 7.3.1, lines 1-2Behaviour of a community

Rationale

The behaviour of a community is defined by a list of elements. We feel that in this list there is an item missing: the one related to force concepts (e.g. delegations).

Proposal

Include a new element in the list:

-- actions due to force concepts (acts, delegations, commitments).

A behaviour of a community is a composition of behaviours that are identified by the roles of the community or a composition of steps in the processes that take place in that community. Constraints on these behaviours must be consistent with the constraints specified by the roles of the community, by the processes and by the relationships between these roles.

FIN44 E7.3.1, page 12, line 4-6 Combinations of policies

Rationale

Unclear text

Proposal

Replace by:

The collective behaviour composed either from role specifications or from process specifications is constrained by the joint policies associated to roles and by the community contract.

Note – there is a substructure in role related policies that can be matched with step constraints.

Editor's Note - The project editor, after redrafting the first sentence of this paragraph according to the instructions in the disposition of comments, feels that the current draft is unclear on the relationship of roles and processes. National Bodies may wish to provide additional text explaining this relationship.

The behaviours of objects in a community are subject to the contract of that community and to the constraints specified in relationships between those objects.

The structure of the community is defined in terms of the following elements:

-- roles;

-- policies for assignment of enterprise objects to roles;

-- relationships between roles;

-- relationships of roles to processes;

-- policies that apply to roles and to relationships between roles;

-- policies that apply to relationships between enterprise objects in the community;

-- behaviour that changes the structure or the members of the community during the lifetime of that community.

AFNOR 15 TH page 11, clause 7.3.1, line 20-21

Rationale

See AFNOR 2

Proposal

Replace by:

-- Changes in the structure or in the members of the community during the lifetime of that community

ES 18 EPage 12, 7.3.1, lines 13-21 Structure of a community

Rationale

The structure of a community is defined in terms of a list of elements. As in our previous comment ES17, we feel that in this list there is an item missing: the one related to force concepts.

Proposal

Include a new element in the list:

-- How and when force actions occur (e.g. acts, delegations, evaluations, commitments).

NOTES

1 – Types of communities or a community template may be used in the specification of a community.

2 – Types of communities may be related by refinement.

3 – A family of related contracts may be generated from a contract template. Some aspects of the contract (e.g. membership) may only apply to particular instantiations of the contract template, while other aspects may apply to all instantiations of the contract template. For example, assignment rules and policies can be considered as parameters in a contract template. The style of contract specification determines the method of community establishment, as well as other aspects of the community life-cycle.

FIN40 TH 7.3.1 Policy framework
Rationale
Use the definition of policy framework in FIN29 to clarify the structure of community specifications.
Proposal
Add to the end of 7.3.1:
Each community must be associated with a single policy framework.
Note - The policy framework may form part of a nested hiearchy of policy frameworks. This may position the community within a larger environment, for example, in respect of some organizations.
Note – The community may be federated, i.e., be composed of domains. The policy framework and policies within it are established when the community is specified or when established according to the specified establishing behaviour. The establishing behaviour may involve domain controllers or already established other communities.

7.3.2 Relationships between communities

FIN2 G Summary of concerns Relationship of communities
Rationale
Federation is not dealt with. Federation means creation of a new community with a new set of governing rules and the federating communities need to be able to provide this specification.
Cluttered otherwise, same conceptual things reappear several times in different terms.
C-object and S-community have a special position, however, it does not clarify the situation as at the same time, it is acknowledged that several c-objects could be simultaneously be interacting within the same community.
Relationships between c-object, role, s-community, refinement via population process are not clear enough.
Relationships between enterprise specifications and organizations are undefined; likewise ownership and domain communities and other community types have relationships that could be clarified.
Community population process is not described.
Proposal
See FIN21, FIN22, FIN24, FIN45, FIN57

FIN45 TH 7.3.2 Relationship between communities

Rationale

The text still gives a number of overlapping examples and fails to reveal all interesting cases. It has been found extremely difficult to interpret.

Proposal

Replacement text:

An enterprise specification can include several communities. The scope of the enterprise specification becomes determined by the choise of one or more root community specifications.

Other communities can further refine the behaviour of these root communities. The communities can be used as a structuring concept, for example, for splitting the specification into subsystems with separate functional responsibilities, into domains for describing ownership and authorization, or into domains for describing controlling and contract making facilities.

The community specifications may overlap each other. In this way it is possible to show how for example functional and administrative communities must interact with each other. The enterprise specification can enforce community interaction at least in the following ways

- communities are nested so that a community becomes a refinement of a role in an outer community and thus inherits all contractual and policy related aspects;

- named roles in one or more communities are required to be fullfilled by the same enterprise object (which can represent a community specified separately, see community object definition in x.x);

- some roles in the community specification are selected and defined so that they explicitly describe how interaction takes place between the communities; more precicely in each community an interface role is described to interact with the community environment and the specifier has to take care that all those environment interactions become captured with interface role actions in appropriate community specifications; and

- one or more community specifications may include behaviour for creating new communities; for example, federation establishment means creation of a new community involving the definition of appropriate policy framework, structure for the community and the community contract.

The actual communities are related to each other following the possibilities created by their specifications. In addition, an enterprise object may fulfil a role in multiple communities as it has not been prevented from doing so. In this case, the amount of information flow or actions involving multiple communities is mainly accidental, and it is the responsibility of the specifier to restrict unwanted interactions.

Enterprise, community and role specifications can be considered as textual items managed by some specification tool and support environment. Reuse of these items is acceptable and the specifications may use textual references to items available in the support environment instead of forcing them to be embedded in the same specification.

An enterprise specification can describe several communities. A community can be considered in the context of some other community or communities to which it is related. These communities may be related in various ways, including relationships when:

AFNOR 17 E page 12, clause 7.3.2, line 32

Rationale

See AFNOR 1

Proposal

Change line 32 to:

An enterprise specification can include several communities.

UK13 Cat TH 7.3.2 Relationships between communities

Rationale:

a) Page 12, line 33-43. This text is tutorial since it is directly derivable from the text on enterprise objects in roles in 7.8.2, Role rules, Page 16 lines17-20, and does not provide structuring rules.

Furthermore, the first, second and fourth bullets can be taken to imply that the fact that the same enterprise object fulfils roles in more than one community necessarily creates a behavioural relationship between those communities. Any such relationship should be explicit in the specification of the communities and roles.

b) Page 13, lines 1-2. This text concerns a relationship between specifications of communities, not the communities themselves. It relates to 7.2, Contents of an enterprise specification, and should be incorporated into that clause.

c) Page 13, lines 3-13. Notes 1 and 4 relate to 7.2, Contents of an enterprise specification, and should be incorporated into that clause.

Note 2 should be stated as a relationship rule.

Note 3 relates to 7.6.1, Establishing a community, and should be incorporated in that clause.

d) Page 13, lines 23-24. This is tutorial text that relates to 7.8.3, Interface roles and interactions between communities, and should be incorporated into that clause.

Proposal: (see UK13<x> above and below for effects at target sites)

a) Move Page 13, lines 1-2, lines 4-5 (Note 1) and lines 11-13 (Note 4) to 7.2 to follow the existing Notes on Page 11, lines 10-15 as additional Notes:

3 – A specification may be partitioned because of readability, reuse of specification fragments in other specifications or interoperability of enterprise objects.

4 – Role and community specifications, as well as type and template specifications, can be private to a specification and development environment, or they can be stored to a repository that can be shared by a wider audience of several development environments and groups.

b) Move Page 13, lines 8-10 (Note 3) to 7.6.1 to follow the existing Note (Page 14, line 36) as Note 2:

2 – The assignment process can be late and dynamic, i.e., a role can be fulfilled by a enterprise object through a match-making process that considers the interacting capabilities, interfaces, behaviour description and, in the

case of a c-object, the policies of the corresponding community with respect to the requirements stated for the role in the other community.

c) Move page 13, lines 23-24 to 7.8.3, as a Note to follow Page 17, line 33.

d) Replace the remainder of the text of 7.3.2 (Page 12, line 32 to Page 13, line 24 with:

An enterprise specification can describe one or more communities. These communities may be related in two ways:

-- a c-object fulfils one or more roles in another community. Under these circumstances, the community that the c-object represents becomes governed by the policy rules of the other community.

-- two c-objects interact in fulfilling roles in another community. Under these circumstances, the communities that the c-objects represent are related by those interactions.

For interaction between two communities to be meaningful, there must be some element of shared objective, which itself implies a higher level of community of which the C-objects representing those communities will both be members, and a common set of policies will apply.

NOTES:

1 – The element of shared objective and the common set of policies can be formed either at design time and included in the specifications of the communities or left for run-time negotiation or more straightforward testing of acceptability during community population.

2 – The communities involved may have differing policy rules all of which the enterprise objects concerned should be able to conform to.

USA 3 G TH Composition of communities

Rationale

The consensus that was reached in Paris on composition of communities has not been carried through in the text.

Proposal

Follow the approach explained by Reenskaug in Paris.

If the working group does not accept this proposal, then the text needs to get more clear and specific about the alternative approach to composition of communities. In particular, the standard needs to clearly specify what it means for an object to participate in an action of a role of a community when that object is not a member of that community.

If there are going to be two approaches to composition of communities, this needs to be made explicit and each needs be described separately.

USA 49 TH Page 12, 7.3.2, lines 32-33 Relationships between communities

Rationale

The meaning of the relationships between communities is defined by the invariants of these relationships. These invariants determine constraints on the collective behaviour of these communities. The list in lines 35-43 provides for important examples.

Proposal

Change: These communities may be related in various ways, including relationships when

To:

The invariants of these relationships between communities determine constraints on the collective behaviour of these communities. This includes relationships when

-- The enterprise object fulfilling a role in one community is part of another community, perhaps fulfilling a role in that community.

-- A composite enterprise object is part of a community and it or one of its component objects is part of another community.

-- A c-object of one community is part of another community, fulfilling a certain role in that community.

USA 50 TH Page 12, 7.3.2, lines 39-40 Relationships between communities

Rationale

This does not correspond the consensus reached at the Paris meeting.

Proposal

From the list headed by: A community can be considered in the context of some other community or communities to which it is related. These communities may be related in various ways, including relationships when:

Remove:

-- A c-object of one community is part of another community, fulfilling a certain role in that community.

USA 51 TL Page 12, 7.3.2, lines 39-40 Relationships between communities

Rationale

This example may be generalized. Such a generalization is recommended since it will demonstrate an important example of a non-binary relationship.

Proposal

Change: A c-object of one community is part of another community, fulfilling a certain role in that community

To:

-- One or more objects of one or several communities (e.g., c-objects or their components) are parts of another community, fulfilling certain roles in a manner prescribed by the invariant of the composition relationship of that community

-- The specification of two or more communities requires a role in each community to be fulfilled by the same enterprise object; since that object participates in the behaviour of each of those communities, it affects the state of all those communities.

-- A portion of the text of the specification of one community is reused in the specification of another community.

USA 52 TL Page 13, lines 1-2 7.3.2 Relationships between communities

Rationale

It is not clear that this example belongs in this list.

We do not suggest that the idea behind this list item should not be included in the draft. However, this does appear to be a different kind of relationship than the others.

Proposal

Remove: -- A portion of the text of the specification of one community is reused in the specification of another community.

NOTES:

1 – Partitioning may be done because of readability, reuse of specification fragments or interoperability of enterprise objects.

USA 53 TL Page 13, 7.3.2, lines 4-5 Relationships between communities

Rationale

The text does not say “partitioning” of what.

Proposal

Clarify or delete: 1 – Partitioning may be done because of readability, reuse of specification fragments or interoperability of enterprise objects.

2 – When a c-object fulfils a role in another community, the community that object represents (a sub-community) becomes governed by the policy rules of the other community.

3 – The population process can be late and dynamic, i.e., a role can be fulfilled by a c-object through a match-making process that considers the sub-community policies, interacting capabilities, interfaces and behaviour description with respect to the requirements stated for the role in the other community.

USA 54 TL Page 13, 7.3.2, lines 6-10 Relationships between communities

Rationale

The objectives and policies of component communities may be mutually inconsistent, and may be inconsistent with the objectives and policies of the composite community. The composition rules should determine how to resolve these inconsistencies.

Proposal

Add:

Note — The objectives and policies of component communities may be mutually inconsistent, and may be inconsistent with the objectives and policies of the composite community. The composition rules determine how to resolve these inconsistencies.

4 - The role and community specifications, as well as type and template specifications, can be private to a specification and development environment, or they can be stored to a repository that can be shared by a wider audience of several development environments and groups.

USA 55 TL Page 13, 7.3.2, lines 11-13 Relationships between communities

Rationale

This Note relates to specification development in general. The rest of the standard does not deal with these issues. It would be preferable to delete the note. If it is decided to retain this note then it should belong elsewhere.

Proposal

Remove: 4 - The role and community specifications, as well as type and template specifications, can be private to a specification and development environment, or they can be stored to a repository that can be shared by a wider audience of several development environments and groups.

For interaction between two communities to be meaningful, there must be some element of shared objective, which itself implies a higher level of community of which both communities will be members, and a common set of policies will apply.

NOTES:

USA 56 TH Page 13, 7.3.2, lines 14-15 Relationships between communities

Rationale

Relationships do not have to be binary

Proposal

Change:

For interaction between ~~two~~ several communities to be meaningful, there must be some element of shared objective, which itself implies a higher level of community of which both these communities will be members, and a common set of policies will apply.

1 – The element of shared objective and the common set of policies can be formed either at design time and included in the specifications of the communities or left for run-time negotiation or more straightforward testing of acceptability during community population.

USA 57 TL Page 13, 7.3.2, lines 18-20 Relationships between communities

Rationale

This statement is not formulated in specification terms

Proposal

Change: 1 – The element of shared objective and the common set of policies can be formed either at design time and included in the specifications of the communities or left for run-time negotiation or more straightforward testing of acceptability during community population.

To:

1 – The common set of policies can be explicitly included in the specification of the communities or can be included as parameter(s) to be instantiated during community population.

2 – The communities involved may have differing policy rules all of which the enterprise object should be able to conform to.

USA 58 TL Page 13, 7.3.2, lines 21-22 Relationships between communities

Rationale

Mutually inconsistent policy rules were referred to earlier, in comments related to lines 6-10 on page 13.

Proposal

Remove: 2 – The communities involved may have differing policy rules all of which the enterprise object should be able to conform to.

Or: reformulate the text in accordance with the proposal regarding lines 6-10

Interactions between enterprise objects fulfilling appropriate roles within different communities can be considered as interactions between those communities.

7.4 Enterprise object rules

An enterprise object is any object in an enterprise specification. The enterprise objects in a specification will be exactly the objects the specifier feels are necessary or desirable to specify the system from the enterprise viewpoint or to understand the enterprise specification.

FIN46 E7.4, lines 26-28, 31-33 community life-cycle

Rationale

Unclear.

Proposal

Replace lines 26-28 with

An enterprise specification may include specifications of enterprise objects; the group of such objects is a free choice of the system specifier.

…

UK14 Cat E Clause 7.4, p13, line 27 Enterprise Object Rules

Rationale

This text includes the sentence: “The enterprise objects in a specification will be *exactly* the objects the specifier feels are necessary or desirable to specify the system from the enterprise viewpoint or to understand the enterprise specification.” (emphasis added). It is not clear what is achieved by the use of the word. It would be strange if the standard recommended vagueness in this so it may be assumed that the word is not there “exactly” in order to prevent any interpretation that the enterprise objects in a specification might be *roughly* the objects the specifier feels are necessary. It is therefore assumed that the intended meaning is that the set of objects included is a modelling choice, driven by the purpose of modelling. If that is the intended meaning the text should be amended as below.

Proposal

Replace the sentence concerned with:

"The set of enterprise objects included in an enterprise specification, and the entities they model, is a modelling decision that will depend on the modelling method being used, and the purpose of developing the model."

NOTE – An enterprise object may be a model of a human being, a legal entity, an information processing system, a resource or a collection or part of any of these.

An enterprise object may be refined as a community at a greater level of detail. All enterprise objects in an enterprise specification fulfil at least one role in at least one community. In fulfilling their roles, enterprise objects, participate in actions, some of which are interactions with other enterprise objects.

FIN46b

…

Replace lines 31-33 with

An enterprise object may be refined as a community at a greater level of detail.

For the purposes of modeling enterprise viewpoint behaviour, the enterprise object behaviour is restricted by the roles it is assigned to.

USA 59 TH Page 13, 7.4, lines 31-33 Enterprise object rules

Rationale

This does not correspond the consensus reached at the Paris meeting.

Proposal

Remove: An enterprise object may be refined as a community at a greater level of detail. All enterprise objects in an enterprise specification fulfil at least one role in at least one community. In fulfilling their roles, enterprise objects participate in actions, some of which are interactions with other enterprise objects.

An enterprise object may be a member of a community because:

-- the community specification provides that the community includes the object,

-- the object is made part of the community at the time of community creation, or

AFNOR 18 E page 13, clause 7.4, line 35

Rationale

See AFNOR 1

Proposal

Change line 35 to:

-- the enterprise specification provides that the community includes the object

-- the object becomes a part of the community as a result of dynamic changes in the configuration of the community.

FIN47 E7.4, last bullet community life-cycle

Rationale

Not precise enough.

Proposal

Replace by

The object joins the community during the life-time of the community.

Note – The community specification includes assignment rules for populating the community, thus, the configuration of identified enterprise objects is not necessary for establishing a community. A role can be empty at times.

Note – The community specification can include rules that change the community structure (for example, number of roles).

7.5 Common community types

UK15 Cat E Clauses 7.5 Common Community Types

Rationale

There is inconsistency in the names used for community types:

a) In Page 13, lines 41-42 and Page 14, lines 7 and 11 the names of the first two community types are:

- domain

- federation

b) In Page 14, lines 8 and 12 the names of the first two community types are:

- <x>-domain

- <x>-federation

c) In Page 13, line 43 the name of the third community type, “ownership community” is inconsistent with the other two.

Proposal

a) Replace Page 13, lines 41-43 with:

- <x>-domain

- <x>-federation

- ownership

b) Replace Page 14, line 7 with:

7.5.1 <X>-domain community type

c) Replace Page 14, line 11 with:

7.5.2 <X>-federation community type

Three community types are:

-- domain

-- federation

-- ownership community

Communities of these types can be specified so that they overlap totally or partially. These basic community types do not imply any hierarchical relationships. A specification may choose to use some or none of these community types.

NOTE: For example, Internet forms a technical domain that is controlled by the standardization organisation responsible for Internet protocols, while the owners of communication connections and computers in Internet each have an ownership community comprising of a set of equipment."

7.5.1 Domain community type

UK14b

b) Replace Page 14, line 7 with:

7.5.1 <X>-domain community type

A community of type <x>-domain contains one "<x>-controller" role and one or more "<x>-controlled" roles, where the controller controls the controlled with regard to the <x>-aspect of their behaviour.

NOTE - The core/environment nature of the controlled role is left to specifiers.

UK16 Cat TL Clauses 7.5.1, Page 14, lines 8-10 Domain Community Type

Rationale

a) Page 14, lines 8-9 introduce undefined terms: "<x>-controller" role, "<x>-controlled" role, "<x>-aspect" and control.

b) The Note, Page 14, line 10, uses the terms "core" and "environment", which are no longer pat of the enterprise language.

Proposal

a) Replace Page 14, lines 8-9 with:

An <x>-domain community comprises an <x>-domain of enterprise objects in the roles of controlled objects and an enterprise object in the role of controlling object for the <x>-domain. The <x>-domain community establishes the characterizing relationship <x> between the enterprise objects in the roles of controlled objects and the enterprise object in the role of controlling object.

b) Delete Page 14, line 10.

7.5.2 Federation community type

UK14c

c) Replace Page 14, line 11 with:

7.5.2 <X>-federation community type

A community of type <x>-federation contains two or more <x>-federation member roles which are filled by <x>-domains. The objective of an <x>-federation is to enable the control of the <x->controlled elements in the individual domains to be shared among the <x>-controllers of those domains. The specific manner in which the <x>-control is shared requires further refinement of the federation community type. <X>-controllers may declare their policies amongst themselves and commit their controlled community to some contract but cannot prescribe policies on each other's controlled communities.

Note. At the level of abstraction at which federation is agreed, the federation members must be domains of the same type (<x>-controlling). However, each <x>-domain may actually be an instance of one or more refined domain types.

UK17 Cat TH Clauses 7.5.2, Page 14, lines 12-19 Federation Community Type

Rationale

a) The UK believes that the intent of this text is to describe a federation of <x>-domain communities as defined in 7.5.1, and not of the <x>-domains alone (thus excluding the controlling objects of the <x>-domains).

Note – The current text is consistent with Part 3, clause 5.1.2, which states: <X> federation – A community of <x> domains. In line with this comment the UK believes that this definition is, also, incorrect and that, if the comment is accepted, a corrigendum should be issued to correct it.

b) The clause includes undefined terms, namely <x>-controlled, <x>-control and <x>-controller.

Proposal

Replace Page 14, lines 12-19 with:

An <x>-federation community ~~of type <x>-federation~~ contains two or more <x>-federation member roles which are filled by <x>-domain communities. The objective of an <x>-federation is to enable the ~~control of the <x->controlled elements in the individual domains among the <x>-controllers of those domains~~ the characterizing relationship <x> between controlled objects and controlling object in each <x>-domain community to be shared with the controlling objects of the other <x>-domain communities. The specific manner in which the ~~<x>-control~~ charactgerizing relationship <x> is shared requires further refinement of the federation community type. ~~<X>-controllers~~ Controlling objects may declare their policies amongst themselves and commit their~~-controlled community~~ <x>-domain communities to some contract but cannot prescribe policies on each other's~~-controlled communities~~ <x>-domain comunity.

Note. At the level of abstraction at which federation is agreed, the federation members must be ~~domains of the same type (<x>-controlling)~~ <x>-domain communities having the same characterizing relationship <x>. However, each <x>-domain community may actually be an instance of one or more refined <x>-domain community types.

Editor's Note - Note that "Generally, the controlling object is not a member of the associated domain" or federation. [2-10.3]

7.5.3 Ownership Community Type

The enterprise object filling the controller role (also known as the owner role) must be the owner of the enterprise objects filling each of the controlled roles (also known as the owned roles). The controlled behaviour is all behaviour of the object filling the owned role apart from that which is controlled in other domains or which cannot be subject to control. The owner may either prescribe policies of the community or delegate an authorization to do so to an agent. This authorization may also be withdrawn.

UK18 Cat E, TL Clauses 7.5.3, Page14, lines 23-27 Ownership Community Type

Rationale

a) The clause does not have the same format as 7.5.1 and 7.5.2.

b) The clause includes undefined terms, namely controlled role, controller role and controlled behaviour.

c) The text implies that an ownership community is a particular type of <x>-domain community. This should be made explicit.

Proposal

Replace Page 14, lines 23-27 with:

An ownership community is an <x>-domain community in which the ~~The~~ enterprise object filling the ~~controller~~ role of controlling object (~~also known as~~ the owner role) must be the owner of the enterprise objects filling each of the ~~controlled~~ roles of controlled object (~~also known as~~ the owned roles). The ~~controlled~~ behaviour that is subject to the characterizing relationship of the <x>-domain is all behaviour of the object filling the owned role apart from that which is subject to the characterizing relationships of ~~controlled in~~ other <x>-domains or which cannot be subject to control. The owner may either prescribe policies of the community or delegate an authorization to do so to an agent. This authorization may also be withdrawn.

USA 60 TH Page 14, 7.5.3, lines 26-27 Enterprise object rules

Rationale

This will not change the meaning but avoids a possible question about the use of authorization here.

Proposal

Change:

The owner may either prescribe policies of the community or delegate ~~an authorization to do so~~ that to an agent. This ~~authorization~~ delegation may also be withdrawn.

Note - Since the owned roles may be filled by enterprise objects of many different types, the extent of control may vary among the owned objects of a common owner object."

7.6 Lifecycle of a community

FIN4 G Summary of concerns Community life-cycle

Rationale

The specified community could exist (be established and have responsibilities) without fixed identified objects populating it – late binding can be used for populating it

Rules for changing the community structure are unclear and unnecessarily strict (no federaition allowed)

How to decide whether an enterprise object can join a community?

– Rules for community to accept a new member as being capable of performing its duties (assignment rules)

– Rules for an object to inherit the policies of the community it joins (inconsistent)

– Rules for an object to check the acceptability of the community policies when joining (missing) or as a preforming member (missing or inconsistent)

Proposal

See FIN39, FIN41, FIN42, FIN46-FIN49

FIN48 TH 7.6 community life-cycle

Rationale

The relationship between the community specification which determines the structure, configuration, rules and policy framework of the community and the community itself is not clear. The text has an underlying assumption that a community is born by instantiating all of its enterprise objects, which in all cases is not wanted. The changes made try to bring out aspects of communtity structures necessary to maintain independently form the enterprise object assignments for the community.

Proposal

7.6.1 Establishing and maintaining a community

A community specification include establishing behaviour for a community. The establishing behaviour may be implicit or explicit, but it establishes the required structures and responsiblities to maintain and control the community, e.g., the policy framework, the community contract and the community membership need to be known.

7.6.2 Populating a community

The community specification includes assignment rules for choosing enterprise objects to fulfill the specified roles. The rules can directly identify the objects wanted, or trust on a supporting mechanism for which more complex rules can be given for population to take place. Such rules may be based on object identities, relationships between objects, object capabilities, technologies, preceding commitments, object behaviour, etc.

The community specified may be a previously existing one, thus the population process only captures the relationships between the objects and the community.

The community specified may need to be created, even, the members of the community may need to be instantiated as the community comes established.

There is no requirement for a community to be fully populated at any time, although the responsiblities taken by the community are present during the life-time of the community. Members to the community can be selected by demand according to the assignment rules for that community. If an enterprise object ceases to fulfill the assignment rule associated to it, it violates the community contract.

7.6.3 Changes in a community

Changes in the structure or behaviour of a community can occur only if the enterprise specification includes behaviour that can cause such changes.

The changes to be considered here include

- introduction of new policy rules within the existing policy framework; or changing the existing policy rule; and

- introduction of new roles into the community.

7.6.4 Terminating a community

An enterprise specification or a community specification may include a terminating behaviour for a community.

Notes

1 – Some communities are permanent and never terminate.

2 – Termination may be triggered by reaching the community objective or failure.

3 - The specification need not to be symmetric in respect to establishing and terminating behaviour. Either of them can be present or missing.

7.6.1 Establishing a community

An enterprise specification can include establishing behaviour for a community.

The establishing behaviour by which a community is established includes the assignment of enterprise objects to roles. The community contract specifies criteria for deciding which objects are so assigned. The enabled behaviour is consistent with the roles.

NOTE – The role/object relationship is not a type/instance relationship.

USA 61 TL Page 14, 7.6.1, line 36 Establishing a community

Rationale

This note may not be needed.

Proposal

Remove:

NOTE – The role/object relationship is not a type/instance relationship.

UK13b

b) Move Page 13, lines 8-10 (Note 3) to 7.6.1 to follow the existing Note (Page 14, line 36) as Note 2:

2 – The assignment process can be late and dynamic, i.e., a role can be fulfilled by a enterprise object through a match-making process that considers the interacting capabilities, interfaces, behaviour description and, in the case of a c-object, the policies of the corresponding community with respect to the requirements stated for the role in the other community.

The enterprise objects assigned to roles in the community can be dynamically changed during the lifetime of the community. As a consequence, a role can, subject to other constraints, be empty. Still, the community is continuously responsible for the obligations placed on that role.

ES 19 EPage 14, 7.6.1, line 35 & 7.6.2, line 41 "empty" roles

Rationale

The word "empty" is used in lines 35 and 41 of page 14 to mean that a role may have no enterprise objects assigned to it. But "empty" could also be read here as "without obligations", that may cause some confusion to the reader.

Proposal (a)

Use "empty" carefully, clarifying its meaning, and try to use "with no enterprise objects assigned to it" instead. Therefore, the sentence "As a consequence, a role can, subject to other constraints, be empty" in line 35 could become:

As a consequence, a role can, subject to other constraints, have no enterprise objects assigned to it.

USA 62 TH Page 14, 7.6.1, lines 37-39 Establishing a community

Rationale

This is repeated in 7.6.2.

Proposal

Remove: The enterprise objects assigned to roles in the community can be dynamically changed during the lifetime of the community. As a consequence, a role can, subject to other constraints, be empty. Still, the community is continuously responsible for the obligations placed on that role.

7.6.2 Changes in a community

Changes in the structure or behaviour of a community can occur only if an enterprise specification includes behaviour that can cause such changes.

AFNOR 19 TH page 14, clause 7.6.2, line 19

Rationale

See AFNOR 2

Proposal

Change:

Changes in the structure or the behaviour of a community can occur only if an enterprise specification includes behaviour that can cause such changes or if the enterprise specification allows changes from entities that are not specified (for example : the designers of the system).

AFNOR 2 G Changes in a community

Rationale

A change in a community can be a creation/deletion of a role, or a creation/deletion of a community, or a modification in a behaviour, or …

We think that these changes can be made either by entities specified by the enterprise specification or by something else (for example the designers). We argue that second case cannot be specified in the enterprise specification.

Proposal

We want to not reduce changes in a community to those that can be specified in the enterprise specification.

USA 63 TL Page 14, 7.6.2, lines 41-42 Changes in a community

Rationale

This paragraph does not consider open systems, the structure or behaviour of which can change due to previously unspecified and unanticipated changes in the environment (due, for example, to a previously unspecified change of epoch). The restriction in this paragraph is unrealistic. It is essential at the very least to recognize that and provide appropriate formulation. (For example, some communities are resilient so that their objectives and important fragments of their invariants tend to be preserved under changes in their environments.)

Proposal

?

Where an enterprise specification allows changes to the assignment of enterprise objects to roles, a role can become empty. In such a case, the specification must ensure that any obligations associated with the role can continue to be fulfilled.

ES 19b EPage 14, 7.6.1, line 35 & 7.6.2, line 41 "empty" roles

Rationale

See ES 19

Proposal b

And the sentence "Where an enterprise specification allows changes to the assignment of enterprise objects to roles, a role can become empty" in lines 40-41 could become:

Where an enterprise specification allows changes to the assignment of enterprise objects to roles, a role can become empty, i.e., with no enterprise objects assigned to it.

7.6.3 Terminating a community

An enterprise specification can include terminating behaviour for a community; it must do so if it includes establishing behaviour for that community.

NOTES – For example, a community contract may provide for termination when the objective is achieved. A violation may be associated with a recovery behaviour, which may be chosen to be the termination of the community.

7.7 Objective rules

TEMPORARY NOTE – The working group invites National Bodies to comment on this clause.

Every community has exactly one objective. This objective is stated in its contract. An objective can be a composition of sub-objectives.

AFNOR 20 TH page 15, clause 7.7

Rationale

If AFNOR 6 is accepted.

Proposal

Add a new line :

Objects and communities may have objectives which influence the choices within some of their behaviour.

JP12 E Page 15, 7.7 lines 11-13 Objective rules

Rationale

Eliminate redundancy.

Proposal

Remove redundant text from the paragraph.

Remove the third sentence from the first paragraph starting “An objective can be …” since the first sentence of the next paragraph is essentially the same.

USA 64 TH Page 15, 7.7, line 11 Objective rules

Rationale

It was agreed at Madrid not to introduce unnecessary differences with Part 3. (Disposition of USA-5)

Proposal

Change:

Every community has exactly one objective. ~~This objective is stated in its contract.~~ The objective is expressed as a contract which specifies how the objective can be met.

USA 65 TH Page 15, 7.7, lines 11-12 Objective rules

Rationale

This text is redundant with the next paragraph.

Proposal

Remove: An objective can be a composition of sub-objectives.

An enterprise specification may decompose the objective of a community into sub-objectives. A sub-objective may be assigned to a role; in that case, the behaviour of the role is specified to meet the sub-objective and the objective is met by the object performing the actions of the role.

USA 66 TH Page 15, 7.7, lines 13-15 Objective rules

Rationale

A sub-objective may be assigned to a collection of roles rather than to a single role (collective behaviour). Also, in accordance with the definition of action in 2-8.3 (“Every action of interest for modelling purposes is associated with at least one object.”), the concept of “actions of the role” ought to be replaced with the concept of “actions in which the role participates”.

Proposal

Change:

An enterprise specification may decompose the objective of a community into sub-objectives. A sub-objective may be assigned to a collection of roles; in that case, the behaviour of the collection of roles is specified to meet the sub-objective and the sub-objective is met by the collection of objects performing the actions of the collection of roles.

A sub-objective may be assigned to a process; in that case, the process is specified to meet the sub-objective and the objective is met by objects performing the actions of the process.

FIN49 TL 7.7 purposeful selection, objective, policy

Rationale

Clarification for the relationship of concepts objective and policy, without the concept of purposeful selection.

Proposal

Line 17, add:

In this case, the sub-objective defines the state in which the process terminates.

Between lines 17-18, add:

The policies of a community restrict the community behaviour in such a way that it is possible to reach the objective. Such policies result in the selection of behaviour that suits the objective of the community.

…

USA 67 EPage 15, 7.7, lines 16-17 Objective rules

Rationale

Clarity. Style

Proposal

Change:

A sub-objective may be assigned to a process; in that case, the process is specified to meet the sub-objective and the sub-objective is met by ~~objects performing~~ the actions of objects performing the process.

Some policies of a community may be criteria for purposeful selection. Such policies result in the selection of behaviour that suits the objective of that community

The objective of an enterprise object is consistent with objectives expressed by the role that it fulfils. Hence, where a c-object fulfils a role in a community, the objective of the community of which the c-object is an abstraction is consistent with the objective expressed by this role.

FIN49b

…

Lines 20-22, replace with:

An enterprise object can only be assigned to a role when the objectives of the object and the community and the role are consistent.

Note - This also applies to community objects and the corresponding more detailed community.

Note – The enterprise specification may provide for detection of conflicts in objectives and for resolution of those conflicts.

USA 68 TH Page 15, 7.7, lines 20-22 Objective rules

Rationale

This does not conform to the consensus reached at Paris.

Proposal

Remove:

Hence, where a c-object fulfils a role in a community, the objective of the community of which the c-object is an abstraction is consistent with the objective expressed by this role.

USA 69 TL Page 15, 7.7, lines 20-22 Objective rules

Rationale

Style. Consistency.

Proposal

Change:

The objective of an enterprise object is consistent with objectives expressed by the role that it fulfils. Hence, where a c-object fulfils a role in a community, the objective of the community of which the c-object is an abstraction is consistent with the objective ~~expressed by~~ assigned to this role.

An enterprise specification may provide for detection of conflicts in objectives and for resolution of those conflicts.

UK19 Cat TH Clause 7.7, Page 15, lines 10-24 Objective rules

Rationale:

a) Page 15, lines 10-24. The current text does not make clear how the objective affects a specification. It talks about objective at two levels of abstraction but does not distinguish between them. These levels of abstraction are:

- the specification itself , where the objective will have constrained design choices in the development of a specification and thus provides information that is necessary for understanding the specification;

- the behaviour defined by the specification, where the objective is a fundamental criterion in any action involving selection between possible behaviours and thus is referenced by such an action.

Text should be provided that distinguishes these levels of abstraction and makes clear how the objective affects the specification.

b) Page 15, lines 16-17. Where sub-objectives are assigned both to processes and to roles, the actions that an object performs in the process must be consistent the sub-objectives of the roles that it fulfils.

c) Page 15, line 20, 1st sentence. This is the only text that talks about the objectives of enterprise objects. There does not appear to be a reason to do so. This text should be deleted.

d) Page 15, line 23-24. Since the objectives (and sub-objectives) in an enterprise specification are defined in the context of that specification they should not be in conflict. This text should be deleted.

e) Page 15, lines 15, 17 and 22. "Objective" should be replaced by "sub-objective".

Proposal:

Replace page 15, lines 10-24 by:

Every community has exactly one objective. This objective is stated in its contract. An objective can be a composition of sub-objectives.

The objective and any sub-objectives apply to the enterprise specification itself, providing the context for the design choices made and enabling understanding.

The objective or a specified sub-objective also applies within the enterprise specification at any point in the behaviour where there is purposeful selection. The purposeful selection results in the selection of behaviour that suits the objective or the specified sub-objective.

~~An enterprise specification may decompose the objective of a community into sub-objectives.~~ A sub-objective may be assigned to a role; in that case, the behaviour of the role is specified to meet the sub-objective and the sub-objective is met by the object performing the actions of the role.

A sub-objective may be assigned to a process; in that case, the process is specified to meet the sub-objective and the sub-objective is met by objects performing the actions of the process. The actions of the process that an object performs are consistent with the sub-objectives of the roles that it fulfils.

~~Some policies of a community may be criteria for purposeful selection. Such policies result in the selection of behaviour that suits the objective of that community.~~

~~The objective of an enterprise object is consistent with objectives expressed by the role that it fulfils. Hence, w~~Where a c-object fulfils a role in a community, the objective of the community of which the c-object is an abstraction is consistent with the sub-objective expressed by this role.

~~An enterprise specification may provide for detection of conflicts in objectives and for resolution of those conflicts.~~

7.8 Behaviour rules

UK20 Cat TH Clause 7.8 Action occurrence

Rationale:

The differences of view on the meaning to be given to the term "action occurrence", as used in ISO/IEC Part 2, 8.3, Note 1, are not yet resolved. While these differences of view do not appear to have affected the text of the EL standard, there is a danger that there could be differences in the interpretation of modelling statements.

The root of the difficulty is that the significance of the terms “action” and “action occurrence” depends on the context in which they are used, for example whether a specification is itself concerned with general behaviour or with an instance of behaviour. The Annex to these comments discusses the kind of issues involved.

In view of the complexity of the issues involved, it is essential to focus on the effects of the issue on the text or use of the standard, and the text necessary to address these specific needs. The UK believes, therefore, that text is required which clarifies the interpretation of the terms in the context of the standard.

Proposal:

a) Insert the following subclause after Page 15, line 25:

7.8.1 Action occurrence

An enterprise specification provides a pattern for an enterprise. As such it may be instantiated once, never, or many times, depending upon the objective of the specifier. This means that the behaviour defined for the enterprise may also be observable any number of times, depending on when and where the specification is instantiated. It is therefore necessary to take care of the context when interpreting statements about the occurrence of the actions that form part of a behaviour.

When distinguishing action type and action occurrence in a specification, the objective is normally to distinguish between multiple occurrences of a single type within the specification, and not to imply a constraint on how often the specification can be instantiated in the real world. The definitions in this section should be interpreted in the context of specification, without constraining when and where the specification should be instantiated.

b) Renumber subsequent subclauses in 7.8

7.8.1 Roles and processes

FIN3 G Summary of concerns Reationship of roles and processes

Rationale

Relationship between roles and processes is unclear.

Proposal

See FIN19, FIN20, FIN51-FIN62

FIN50 – see Clause 5

TEMPORARY NOTE – The working group notes that collective behaviour needs to be considered here

The behaviour of a community is a composition of the expected actions in which the objects of the community participate in fulfilling the roles of the community, together with a set of constraints on when these actions may occur. These constraints may define the possible ordering of these actions and the decomposition of some actions.

USA 71 TH Page 15, 7.8.1, lines 28-33 Roles and processes

Rationale

Style. Consistency.

Proposal

Change:

The behaviour of a community is a composition of the ~~expected~~ actions in which the objects of the community participate in fulfilling the roles of the community, together with a set of constraints on when these actions may occur. These constraints may define the possible ordering of these actions and the decomposition of some actions.

NOTE – There are many specification styles for expressing ~~the ordering of~~ when actions may occur. The modelling language chosen for expressing an enterprise specification may impose certain styles.

NOTE – There are many specification styles for expressing the ordering of actions. The modelling language chosen for expressing an enterprise specification may impose certain styles.

The assignment of actions to the enterprise objects that comprise a community is defined in terms of roles. A role identifies an abstraction of the community behaviour. All of the actions of that abstraction 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 abstraction is labelled as a role. The behaviour identified by that role is subject to the constraints specified in the contract of the community and in the structure of the community. The emphasis is on the enterprise objects that participate in the particular behaviour.

UK21 Cat E Clause 7.8.1 Roles and processes

Rationale:

Clarification.

Proposal:

a) Page 15, lines 39-40. Replace the last sentence by:

In contrast to the specification of actions and their ordering in terms of processes (see below), the emphasis is on the enterprise objects that participate in the particular behaviour.

b) Page 16, lines 9-10. Replace the last sentence by:

In contrast to the specification of actions as related to roles (see above), the emphasis is on what the behaviour achieves.

USA 72 TH Page 15, 7.8.1, lines 34-40 Roles and processes

Rationale

Style. 'Structure of the community' is not defined.

Proposal

Change:

The assignment of actions to the enterprise objects that comprise a community is defined in terms of roles. A role identifies an abstraction of the community behaviour. All of the actions of that abstraction 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 those abstractions is labelled as a role. The behaviour identified by that role is subject to the constraints specified in the contract of the community and in the ~~structure~~

configuration of the community. The emphasis is on the enterprise objects that participate in the particular behaviour.

USA 70 TH Page 15, 7.8.1, lines 27, 35 Roles and processes

Rationale

Since collective behaviour has to be considered in a consistent manner, the concepts of "action of an object" or "action of a role" ought to be replaced with the concepts of "actions in which an object participates" or "actions in which a role participates".

Comment. Can we say that an interaction (of several objects) is a way to represent collective behaviour?

Proposal

Change:

All of the actions ~~of that~~ in which this abstraction participates.

Role behaviour decomposes the behaviour of the community into roles that can each be performed by an enterprise object in the community. The enterprise object that performs the role behaviour is said to fulfil that role within the community or is said to be assigned to that role within the community.

USA 73 E Page 16, 7.8.1, lines 1-2 Roles and processes

Rationale

Style

Proposal

Change:

~~Role behaviour~~ Roles are used to decomposes the behaviour of the community into roles that can each be performed by an enterprise object in the community

Each action will be part of at least one role behaviour, but can be part of many role behaviours (when the action involves an interaction).

Editor's note – Or involves collective behaviour.

The actions and their ordering can be defined in terms of processes. A process identifies an abstraction of the community behaviour that includes only those actions that are related to achieving some particular result/purpose/sub-objective within the community. Each abstraction is labelled with a process name. The emphasis is on what the behaviour achieves.

UK21b

b) Page 16, lines 9-10. Replace the last sentence by:

In contrast to the specification of actions as related to roles (see above), the emphasis is on what the behaviour achieves.

Process behaviour decomposes the behaviour of the community into processes.

Note – The choice of using a role-based or process-based modelling approach will depend on the modelling method used and the aim of modelling. It may be necessary to use a combination of the two approaches.

JP13 E Page 16, 7.8.1 lines 1-13 Roles and processes

Rationale

The terms “role behaviour” and “process behaviour” are misleading.

Proposal

Rename those as follows.

Rename “role behaviour” (line 1 and 2) by “role by behaviour,” and “process behaviour” (line 11) by “process by behaviour.”

7.8.2 Role rules

In the specification of a community, each role stands as a placeholder for some enterprise object that exhibits the behaviour identified by the role.

FIN55 E7.8.2, page 16, lines 16, 39-41 role rules

Proposal

Add on line 16:

Integrally related to the specification of a role there is an assignment rule that sets requirements for the potential objects to fulfil that role.

…

AFNOR 21 E page 16, clause 7.8.2, Line 15

Rationale

See AFNOR 1

Proposal

Change line 15-16 to:

In a community, each role stands as a placeholder […] by the role.

An enterprise object may fulfil several roles in one community, and may fulfil roles in several communities. A role in a community may be filled by different objects at different times, but only by one enterprise object at any one time. At any location in time a role in a community may be unfilled, provided that the constraints in the specification of the community so permit.

AFNOR 22 E page 16, clause 7.8.2, Line 19-20

Rationale

See AFNOR 1

Proposal

Change line 19-20 to:

At any location […] that the constraints in the community so permit.

ES 20 EPage 16, 7.8.2, lines 12-15 Role rules

Rationale

It is explicitly stated in page 16, lines 12-15, that "A role in a community may be filled by different objects at different times, but only by one enterprise object at any one time". However, a c-object may be a possible enterprise object, and therefore a community can fill a role in another community. This is not explicitly mentioned, and we think that it should be.

Proposal

Include a note after the paragraph in lines 12-15:

NOTE - A c-object may be a possible enterprise object, and therefore a community can fill a role in another community.

An enterprise specification may include a number of roles of the same type each fulfilled by distinct enterprise objects, possibly with a constraint on the number of roles of that type that can occur.

FIN52 E7.8.2, lines 21-23, line30 role rules

Proposal

Move lines 21-12 just before line 30.

NOTE - Examples are modelling the members of a committee and the modelling of the customers for a service.

When a community template is instantiated, at most one enterprise object is associated with each role. The constraints of the behaviour named by the role become constraints on the object fulfilling the role.

AFNOR 23 TH page 16, clause 7.8.2, Line 24-25

Rationale

It seems that there is a contradiction. Lines 19-20 express that a role may be unfilled but line 24-25 express the fact that an enterprise object is associated with each role. We think that when a community template is instantiated some role can be unfilled.

Moreover the second sentence is not reduced to instantiation of template.

Proposal

Remove line 24. Change in line 24-25 (the second sentence) "behaviour named by the roles" by "behaviour identified by the roles".

Editor's Note - These constraints also become constraints on other objects later fulfilling that role. If this is implied by other role rules, then the previous sentence may not be necessary.

NOTE – An enterprise object may come to fulfill a role in a community in other ways than by the instantiation of a template (see subclause 7.1).

FIN53 E7.8.2, lines 17-20 and 24-29 role rules

Rationale

Contradictory text.

Proposal

Replacement text:

An enterprise object may fulfil several roles in one community, and may fulfil roles in several communities. An object associated to several roles becomes constrained simultaneously by all the behaviours named by the roles and the policies associated to these roles.

At any location in time at most one enterprise object is associated with each role. The constraints of the behaviour named by the role become constraints on the object fulfilling the role. A role may be filled by different objects at different times, even, a role may be unfilled, provided that the specification of the community so permit.

When an enterprise object is assigned to a role the types of the role should not contradict any of the types satisfied by that object unless the specification includes mechanisms to determine and resolve such inconsistencies.

FIN54 E7.8.2, line 30 role rules

Rationale

Unclear.

Proposal

Replace by

Enterprise objects of different types may be assigned to a role, given that the object type not contradicts with the role type.

Note - The community supporting system environment or the community specification may include mechanisms to determine and resolve inconsistencies between object types and requirements set by roles, thus enlargening the group of acceptable objects for a given role.

An enterprise sepecification may allow roles and relationships between roles to be created or deleted during the lifetime of the community. The role lifetime is contained within the community lifetime, and the period for which a particular enterprise object fulfils a given role is contained within the lifetime of that role.

NOTE - The constraints of the community should be satisfied throughout its lifetime. However, these invariants may change; this may determine different epochs in this lifetime. Such changes may lead to changes in the sets of roles and in the sets of relationships between roles of the community.

The policies of a community apply to the behaviour of the objects in the community and to the structure of the community. Policies of the community may include prescriptions concerning the assignment of enterprise objects to roles.

FIN55b E7.8.2, page 16, lines 16, 39-41 role rules

Proposal

…

Replace 39-41:

Assignment rules for a role include prescription concerning the assignment of enterprise objects to roles.

Note - The existence of these rules in a community specification is compulsory, but the form in which they appear is dependent on the support mechanisms for that community. Some execution environments or specification tools support only assignment of identified objects, others support type based matching and late binding within the communities.

NOTES

1 – The concept of role de-couples the expected behaviour from the identities of particular enterprise objects. A role is a placeholder (a formal parameter) providing an identifier for some part of the community behaviour. Associated with a role is a set of constraints on the behaviour expected of any enterprise object that is to fulfil the role; these are requirements on the candidate types of enterprise object.

A focus in describing a community is on the consequences of the community being in existence, as well as on the interaction of its members. What emerges from the community behaviour is a series of constraints on the behaviour of those members – a set of community policies.

2 –If the term '<x> object' is used in an enterprise specification, where <x> is a role, it should be interpreted as meaning 'an enterprise object fulfilling the role, <x>'. Where an enterprise object fulfils multiple roles, the names can be concatenated.

Editor's Note - The following two paragraphs were in subclause 7.7.2.2 of the second committee draft.

TEMPORARY NOTE – There is no consensus in the working group whether an enterprise specification can represent actions that are not in a behaviour identified by a role.

NOTE – The structuring rules in Part 3 prescribe that an enterprise specification starts with a specification of a community that consists of the ODP system being specified and the objects in its environment. There are no objects in the specification outside this community.

FIN56 E7.8.2, page 16, line 39 – page 17, line 15 role rules

Rationale

Repetitive, and at least partially incorrect.

Proposal

Remove notes; except note 2 that may be kept and moved under line 29 on page 16.

7.8.3 Interface roles and interactions between communities

FIN57 E7.8.3 Interactions between communities

Rationale

Contents already revealed by 6.2.7 more clearly (second case), first case has nothing to add for relationships between communities (not actually talking about interface roles but interactions between roles within an additional community).

Proposal

Delete clause.

USA 75 TH Page 17, 7.8.3, lines 16-33 Interface roles and interactions between communities

Rationale

The consensus that was reached in Paris on composition of communities has not been carried through in the text.

Proposal

We propose that an e-mail meeting be started to work out an explanation of interface roles and interactions between communities that corresponds to the consensus reached in Paris. At the same time, language can be worked out that more clearly explains the approach of this text.

Objects of a community may interact with objects outside that community to achieve its objectives. This can occur in one of two ways:

-- either an object outside the community and an object inside the community interact as part of another community;

USA 74 TH Page 17, 7.8.3, lines 19-20 Interface roles and interactions between communities

Rationale

Relationships do not have to be binary

Proposal

Change:

either ~~an~~ a non-empty set of objects outside the community and ~~an~~ a non-empty set of objects inside the community interact as part of another community

-- or the objects outside the community interact with an object (a c-object) that is a composition of some or all of the objects of the community. In this case there are interface roles in the community that identify these interactions.

The second case occurs when a community is specified at two different levels of abstraction:

-- as the configuration of enterprise objects that is the community, and

-- as a c-object that is an abstraction of the community and is part of a community of wider scope. Then actions in which the c-object participates are identified by interface roles of the community.

Temporary Note – In the first case, is there a need to state any constraints that apply between the role of an object in a community and the role of the same object in a second community in which it interacts with an object outside the first community?

UK22 Cat TH 7.8.3 Page 17, lines 17-31 Interaction roles and interactions between communities

Rationale:

Interactions of an enterprise object in a community that are relevant to that community only occur in fulfilling a role of that community. Thus, the interactions referred to in Page 17, lines 19-20 are not relevant to the community referred to in Page 17, line 17 and lines 21-23, and should not be considered in the context of that community.

Proposal:

Revise Page 17, lines 17-31 as follows:

A role in ~~Objects of~~ a community may identify behaviour that includes interactions ~~interact~~ with objects outside that community to achieve the objective of the community~~its objectives. This can occur in one of two ways:~~

~~-- either an object outside the community and an object inside the community interact as part of another community;~~

~~-- or the objects outside the community interact with an object (a c-object) that is a composition of some or all of the objects of the community. In this case there are interface roles in the community that identify these interactions.~~

~~The second case occurs when~~ In such a case a community is specified at two different levels of abstraction:

-- as the configuration of enterprise objects that is the community, and

-- as a c-object that is an abstraction of the community and is part of a community of wider scope. ~~Then~~ actions in which the c-object participates are identified by interface roles of the community.

~~Temporary Note – In the first case, is there a need to state any constraints that apply between the role of an object in a community and the role of the same object in a second community in which it interacts with an object outside the first community?~~

If the community is represented as a c-object, a mapping should be specified between the interface of the c-object and the interface roles in actual specifications.

UK23 Cat E Clause 7.8.3 Interface roles and interactions between communities

Rationale:

Clarification and correction.

Proposal:

Page 17, lines 32-33. Make the following changes:

If the community is represented as a c-object, a mapping should be specified between each interaction~~the interface~~ of the c-object and an interaction of an interface role of the community~~the interface roles in actual specifications~~.

UK13c

c) Move page 13, lines 23-24 to 7.8.3, as a Note to follow Page 17, line 33.

Note: Interactions between enterprise objects fulfilling appropriate roles within different communities can be considered as interactions between those communities.

7.8.4 Enterprise objects and actions

FIN58 E7.8.4 Enterprise objects

Rationale

Clause 7.8.4 gives only trivial rephrasing of clause 6 definitions. All rules related to enterprise objects are given through roles in the specifications, and thus become stated in 7.8.2.

Proposal

Remove 7.8.4. Rename 7.8.2 as Role and enterprise object rules.

An enterprise object fulfils at least one role in at least one community and can be involved in actions in the following ways:

-- The object can participate in carrying out the action; in this case it is said to be an actor with respect to that action.

-- The object can be mentioned in the action; in this case it is said to be an artefact with respect to that action.

-- The object can both be essential for the action and require allocation or possibly become unavailable; in this case it is said to be a resource with respect to that action.

NOTE – For every action there is at least one participating enterprise object. Where two or more enterprise objects participate in an action, it is an interaction. When only one enterprise object participates in an action, it may be an interaction, if the object interacts with itself. [2-8.3]

USA 76 TH Page 17, 7.8.4, lines 35-45 Enterprise objects and actions

Rationale

The text repeats what has already been said.

Proposal

Remove:

An enterprise object fulfils at least one role in at least one community. ~~and can be involved in actions in the following ways:~~

~~-- The object can participate in carrying out the action; in this case it is said to be an actor with respect to that action.~~

~~-- The object can be mentioned in the action; in this case it is said to be an artefact with respect to that action.~~

~~-- The object can both be essential for the action and require allocation or possibly become unavailable; in this case it is said to be a resource with respect to that action.~~

NOTE – For every action there is at least one participating enterprise object. Where two or more enterprise objects participate in an action, it is an interaction. When only one enterprise object participates in an action, it may be an interaction, if the object interacts with itself. [2-8.3]

In the special case where an enterprise object is mentioned in an action in which it also participates (e.g. an enterprise object reporting its state) it is both an actor and an artefact with respect to that action.

USA 77 TH Page 18, 7.8.4, lines 1-2 Enterprise objects and actions

Rationale

We do not understand the example. We expect it will confuse others.

Proposal

Change:

In the special case where an enterprise object is mentioned in an action in which it also participates ~~(e.g. an enterprise object reporting its state)~~ it is both an actor and an artefact with respect to that action.

In the special case where an enterprise object is used in an action in which it also participates it is both an actor and a resource with respect to that action.

ES21 E page 17,lines 37-42 actor&artefact, actor&resource

Rationale

The wording of lines 37-40 makes very difficult its understanding, and may be even incorrect in some parts. (as the editor points out in his Note)

Proposal

Re-write those two paragraphs, making use of the previous definitions of actor, artefact and resource. This could lead to the (more simplified) definition, with two sentences that comprise both paragraphs:

An actor in an action can also be an artefact with respect to that action (e.g., an enterprise object reporting its state). Likewise, an actor in an action can also be a resource with respect to that action (if it itself is used in the action)

Editor's Note – This is problematical. A resource used in an action certainly participates in that action

When a resource is essential for some action, the action is constrained by the availability of that resource.

Where a role in a community involves an enterprise object in actions only as an artefact, then it is an artefact role in that community.

Where a role in a community involves an enterprise object in any action as a resource, then it is a resource role in that community.

Where a role in a community involves an enterprise object in any action as an actor, then it is an actor role in that community.

USA 78 TH Page 18, 7.8.4, lines 8-13 Enterprise objects and actions

Rationale

The text repeats what has already been said.

Proposal

Remove: Where a role in a community involves an enterprise object in actions only as an artefact, then it is an artefact role in that community.

Where a role in a community involves an enterprise object in any action as a resource, then it is a resource role in that community.

Where a role in a community involves an enterprise object in any action as an actor, then it is an actor role in that community.

NOTES

1 – Therefore, roles in a community can be partitioned into actor roles, artefact roles and resource roles with respect to that community.

USA 79 TH Page 18, 7.8.4, lines 15-16 Enterprise objects and actions

Rationale

The note is false. (Line 4 states that a role may involve an enterprise object both as a resource and as an actor even in the same action.)

Proposal

Remove:

1 – Therefore, roles in a community can be partitioned into actor roles, artefact roles and resource roles with respect to that community.

2 – A role is specified in such a way that the behaviour associated with the role, the policies applying to the role, the responsibilities associated with the role, and the relationships with other roles are stated. For example, for each actor role there are complete descriptions of all actions of objects fulfilling that role, and for each action, identification of all the artefact roles mentioned in the action.

ES22 E page 18, Note 2, lines 10-16 Editor's notes

Rationale

The second sentence in Note 2 may be too restrictive in many situations, as the editor's notes suggest.

Proposal

Re-write Note 2, allowing for some abstract specifications that could be refined at a later stage.

2 – A role is specified in such a way that the behaviour associated with the role, the policies applying to the role, the responsibilities associated with the role, and the relationships with other roles are stated. For example, for each actor role there are descriptions of the actions of objects fulfilling that role, and for each action, identification of artefact roles mentioned in the action, and eventually the resource roles that any object fulfilling that role may use.

UK24 Cat TH 7.8.4, p18 lines 8-23 Actor, Artefact and Resource Roles

Rationale:

See UK9.

Proposal:

Delete p18 lines 8-13 and 15-16; amend lines 17-20 (the Note) as follows:

A role is specified in such a way that the behaviour associated with the role, the policies applying to the role, the responsibilities associated with the role, and the relationships with other roles are stated. For example, for each role there are complete descriptions of all actions of enterprise objects fulfilling that role, and for each action, identification of all the artefacts mentioned in the action.

Editor's Notes

1 – 'Complete' is ambiguous in this context.

2 – Might the specification of an actor role be abstract, hiding some details.

USA 80 TL Page 18, 7.8.4, lines 17-23 Enterprise objects and actions

Rationale

The text should take into account collective behaviour. Specifically, it is not necessary to specify behaviour, policies and relationships in which a role participates in the specification of that role. Such a prescription imposes unnecessary modelling restrictions and choices such as the need to determine the specific role in the specification of which the collective behaviour of several roles will be included ("owner of the behaviour"). Also, editor's notes ought to be taken into account.

Proposal

Change:

2 – ~~A~~ Roles is specified in such a way that the behaviour associated with the roles, the policies applying to the roles, the responsibilities associated with the roles, and the relationships ~~with other~~ between roles are stated. For example, for each actor role ~~there are complete descriptions of all actions of objects fulfilling that role~~ the specification includes descriptions of all actions in which the objects fulfilling that role participate, and for each action, ~~identification of all the artefact roles mentioned in the action~~ the specification includes descriptions of all the artefact and resource roles mentioned in the action.

USA 81 TL Page 18, 7.8.4, lines 17-20 Enterprise objects and actions

Rationale

The text repeats what is said elsewhere.

Proposal

Remove:

2 – A role is specified in such a way that the behaviour associated with the role, the policies applying to the role, the responsibilities associated with the role, and the relationships with other roles are stated. For example, for each actor role there are complete descriptions of all actions of objects fulfilling that role, and for each action, identification of all the artefact roles mentioned in the action.

USA 82 TL Page 18, 7.8.4, lines 17-20 Enterprise objects and actions

Rationale

If the previous comment is not accepted, make the prescriptive part normative. Remove 'complete.'

Proposal

Change:

The specification of a role ~~is specified in such a way that~~ includes the behaviour associated with that role, the policies applying to that role, the responsibilities associated with that role, and the relationships with other roles ~~are stated~~.

NOTE- For example, for each actor role ~~there are complete descriptions of~~ all actions of objects fulfilling that role are specified, and for each action, ~~identification of~~ all the artefact roles mentioned in the action are identified.

7.8.5 Process rules

FIN59 TH 7.8.5 Process rules

Rationale

Current text causes more questions than it is able to give answers, mainly because of inconsistencies.

Proposal

Replace by

In an enterprise specification, a process is an abstraction of the behaviour of some configuration of objects in which the identities of objects have been hidden as a result of the abstraction. Each step need not be performed by the same enterprise object. A step can even be performed jointly by multiple roles.

If processes are used as part of the community specification, each step must be associated to an actor role.

The process specification must include specification of the state which initiates the process, and the state that terminates the process.

The collective behaviour of a community is a set of processes. This set can be seen as a more abstract process performed by a single role performed by a c-object. Also, a step of a process can be further refined as a more detailed process.

7. 8.5.1 Specifying processes

A process is a directed acyclic graph of steps, in which each step is an action that:

-- is made possible by state changes of one or more enterprise objects, or the completion of one or more previous steps in the process

-- results in state changes of one or more enterprise objects, some of which make possible the occurrence of subsequent steps in the same process.

A process is initiated by the occurrence of specified state.

A process is terminated by the entry into a specified state.

Editor's Note – The bullet points in the first paragraph of this subclause suggests that the second and third paragraphs should end with 'specified state or states.'

Note that a state is always the state of an object.

NOTES –

1 – The use of 'acyclic' indicates that the trace, or history, of steps does not contain cycles of cause and effect. This does not prevent the use of notations with a concept of iteration; such looping concepts generate a sequence of distinct step occurrences.

2 – In an enterprise specification, a process is an abstraction of the behaviour of some configuration of objects in which the identities of the some objects have been hidden as a result of the abstraction.

Each step need not be performed by the same enterprise object in the community. A step in a process may itself be modelled as a (sub-)process.

USA 84 TH Page 18, 7.8.5.1, line 42 Specifying processes

Rationale

Collective behaviour should be taken into account. A step need not be performed by a single object; "performance" in this context is a concept from the message-oriented object model.

Proposal

Change:

Each step need not be ~~performed by the same enterprise object in the community~~ an abstraction of the behaviour of the same configuration of objects in the community.

7. 8.5.2 Mapping between processes and roles

In any mapping between processes and roles each step identified in a process is associated with at least one actor role. Different steps in a process can be associated with different actor roles.

FIN51 E7.8.1, 7.8.5.2 process

Rationale

After FIN19 clause 7.8.1 is empty and the text does not really need to be in a separate clause.

Proposal

Move 7.8.5.2 to form a new version of 7.8.1

7.9 Policy rules

FIN5 G Summary of concerns Behaviour and policies

Rationale

Situations may arise in which an object is allowed to ask for a service which is prohibited from use. No failure or violation management concepts are offered.

Relationship between policy and behaviour is unclear.

Relationship between processes and policy are unclear.

Proposal

See FIN14-FIN18, FIN27-FIN31, FIN40, FIN44, FIN50, FIN63-FIN67

A policy specification is an identified part of an enterprise specification that is likely to evolve during the lifetime of the ODP system or that can be changed to tailor a single specification to apply to a range of different ODP systems.

A policy specification includes:

-- the name of the policy;

-- the rules;

-- the elements of the specification affected by the policy;

-- constraints on changing the rules;

-- rules for changing the policy.

ES23 E page 19, 7.9, lines 5-9 Editor's note on rules for changing policies

Rationale

As pointed out by the editor, in the list of elements that a policy specification includes, one of the last two items can be eliminated. More precisely, “constraints for changing the rules” can be viewed as a part of “rules for changing the policies”. Here, “rules” is more general than “constraints”, and “policies” is more general than “rules”, which implies that the second sentence is more general (and includes) the first one.

Proposal

Remove the item “constraints for changing the rules”

Editor's Note – Are the last two items the same? The policy is the set of rules. “Policy: A set of rules related to a particular purpose.” [2-11.2.7] (Some) rules are constraints. “Not every policy is a constraint.” [2-11.2.7]

FIN60 E7.9, page 19, lines 13-16 policy rules

Rationale

Answer to editor question on line 14: No.

Proposal

Change line 13:

behaviour for changing the policy

USA 83 TH Page 18, 7.9, lines 12-16 Policy rules

Rationale

The last two items are the same. (Or, if different, the meanings are not clear.)

Proposal

Remove:

-- constraints on changing the rules;

Leave:

-- rules for changing the policy.

USA 85 TH Page 19, 7.9, lines 12-16 Policy rules

Rationale

The last two items are the same. (Or, if different, the meanings are not clear.)

Proposal

Remove:

-- constraints on changing the rules;

Leave:

-- rules for changing the policy.

Changes in the policies of a community can occur only if an enterprise specification includes behaviour that can cause such changes.

Note – Policy may, for example, be used to configure generic components to apply them in some specific situation, or to express a pervasive decision that affects many components.

ES24 TL Page 19, 7.9 Policy rules

Rationale

Here we are talking about the things that may change. Can we change the system objectives (section 7.7), too? If so, how to specify the evolution of the system objectives and sub-objectives?

Proposal

If a system's objectives can change, there should be a policy specification stating which ones may change, when they can change, who can change them, and how.

We propose including a Note in the text about this.

7.9.1 General policy rules

FIN61 TH 7.9.1 policy rules

Proposal

Replace by:

In an enterprise specification, policies can be associated with

- communities, including the root community representing whole ODP systems
- roles
- steps as parts of processes
- enterprise objects.

Include 7.9.5 here

Include 7.9.3 here

A community specification can include policies that apply to the community as a whole.

JP14 E Page 19, 7.9.1 lines 22-23General policy rule

Rationale

"Community specification" is used without clear definition or explanation. This term also appears in NOTE 4 of 7.3.2 and in 7.4.

Proposal

Add explanation of this term before using it.

Add, at the end of 7.3.1, a specification of community is referred to as community specification.

The policies specified in a community specification are directly related to the achievement of the objective of the community.

As a result of the creation of a community the policies that apply to the enterprise objects that are its members can be changed in the following ways:

-- the enterprise objects, by being subject to the agreement which creates the community, are subject to its policies and their behaviour is subject to different constraints from those that applied before;

-- the enterprise objects may be subject to obligations not to do things unless specifically allowed by the rules of the community;

7.9.2 Obligations, permissions and prohibitions

Policies may apply to enterprise objects (in all roles), roles (for all actions named by a role or set of roles), or to an action type or set of action types named by a role or set of roles. They may also apply to the collective behaviour of a set of enterprise objects

FIN62 E7.9.2 policy rules

Proposal

Delete lines 33-35, as they cause more confusion than clarification.

UK25 Cat E Clause 7.9 Introduction, 7.9.1 and 7.9.2 Policy rules

Rationale:

a) Page 19, line 5 to page 20 line 33. This text is about the specification of policy and this should be refelected in the subheadings.

b) Page 19, line 5 and line 8. Use of the term "policy specification" is not in line with usage elsewhere in the document.

c) Page 19, lines 5-6. A policy may be changed; it will not evolve.

d) Page 19, line 10. It should be stated that the rules are expressed as obligations, permissions and obligations.

e) Page 19, line 13. This should be expressed as "behaviour for changing the policy" and allow for the case when the policy cannot be changed.

f) Page 19, lines 17-20. This should come after lines 5-7.

g) Page 19, lines 21-31:

(i) The sub heading is not required.

(ii) Line 22 should be merged with page19, lines 33-35.

(iii) Lines 23-24 is covered by 7.7, Objective rules.

(iv) Lines 25-31 are tutorial and should be made into a Note.

h) Page 19, lines 33-35. This text should be moved to follow page 19, line 7.

Proposal:

Replace Page 19, lines 5-35 by:

7.9 Policy rules

7.9.1 The specification of a policy

A policy identifies the specification of a behaviour, or constraints on a behaviour, that can be changed during the lifetime of the ODP system or that can be changed to tailor a single specification to apply to a range of different ODP systems. Changes in the policies of a community during its lifetime can occur only if an enterprise specification includes behaviour that can cause such changes.

NOTES:

1 – A policy is a named place-holder for a piece of behaviour used to parameterize a specification in order to facilitate response to later changes in circumstances. The behaviour of systems satisfying the specification can be modified by changing the policy value, subject to constraints associated with the policy in the original specification. In these terms, a policy is an aspect of the specification that can be changed, and a policy value is the choice in force at any particular instant. Thus one might speak of a scheduling policy with a FIFO policy value.

2 – Policy may, for example, be used to configure generic components to apply them in some specific situation, or to express a pervasive decision that affects many components.

Policies may apply to a community as a whole, to enterprise objects (in all roles), to roles (for all actions named by a role or set of roles), or to an action type or set of action types named by a role or set of roles. They may also apply to the collective behaviour of a set of enterprise objects

The specification of a policy includes:

-- the name of the policy;

-- the rules, expressed as obligations, permissions and obligations;

-- the elements of the enterprise specification affected by the policy;

-- constraints on changing the rules;

-- behaviour for changing the policy (which may be null, i.e. the policy is not changed during the lifetime of the community).

7.9.2 The specification of obligations, permissions and prohibitions

7.9.2.1 Obligation

An obligation is defined etc.

7.9.2.1 Obligation

An obligation is defined by:

-- a behaviour;

-- a role or roles involved in that behaviour;

-- a predicate on behaviour;

-- an authority that imposes the obligation.

If a set of enterprise objects is subject to the obligation then, when the predicate is true, there is an obligation on the set of objects to participate in the behaviour, by order of the authority.

A standing obligation is an obligation for which the predicate on the behaviour is always true

7.9.2.2 Permission

A permission is defined by:

-- a behaviour

-- a role or roles involved in that behaviour

-- a predicate on behaviour

-- an authority which grants the permission

If a set of enterprise objects has this permission, then, when the predicate is true, that set of enterprise objects is allowed to take part in the action when fulfilling the roles, by order of the authority.

Editor's Note – Does 'allowed' convey the meaning of permission? The following may be more precise:

If a set of enterprise objects has this permission, then, when the predicate is true, there is no obligation for that set of enterprise objects not to take part in the action when fulfilling the roles, by order of the authority.

FIN63 TH 7.9.2.2 permission and successful operation

Proposal

Replace editor note by

Note – There is, however, no guarantee that the action succeeds. The action may have participants, for example, at other domains on which the action is prohibited.

ES25 E page 20, 7.9.2.2, lines 6-10 Editor's note on permission

Rationale

The editor's proposed text in his Note contains a double negation, which makes the proposed sentence very obscure and difficult to understand.

Moreover, we think that "allow" was correctly used in the text, and that it conveys the required meaning of permission.

cf. OXFORD dictionary:

allow *v*. **1** *tr*. permit (a practice, a person to do something, a thing to happen, etc.) ....

Proposal

Remove the editor's note.

7.9.2.3 Prohibition

Like a permission, a prohibition is defined by:

-- a behaviour

-- a role or roles involved in that behaviour

--a predicate on behaviour

-- an authority which imposes the prohibition

If a set of enterprise objects has this prohibition, then, when the predicate is true, that set of enterprise objects cannot take part in the action when filling the roles, by order of the authority.

FIN64 E7.9.2.3prohibition

Rationale

Clarification.

Proposal

Line 20: delete "like a permission".

Line 26: Cannot -> is not allowed.

ES26 E page 20, 7.9.2.3, lines 19-22 Editor's note on prohibition

Rationale

Again, we think that the editor's proposal to avoid "cannot" is difficult to understand.

Proposal

Replace "cannot" with "are not allowed to" in line 18.

If a set of enterprise objects has this prohibition, then, when the predicate is true, that set of enterprise objects are not allowed to take part in the action when filling the roles, by order of the authority.

Editor's Note – Might 'cannot' be too strong here? The following text may be more precise:

If a set of enterprise objects has this prohibition, then, when the predicate is true, there is an obligation for that set of enterprise objects not to take part in the action when fulfilling the roles, by order of the authority.

Editor's Note - Is a subclause wanted here, prescribing the form to be used to specify an authorization?

FIN65 TL 7.9.2.3, editor question authorization

Rationale

Proposal

No need for authorization subclause. Check spelling of authorization/authorisation.

ES27 E page 20, 7.9.2, lines 24-25 Editor's note on Authorisation

Rationale

We agree with the editor: a 7.9.2.4 section is needed to deal with "authorisation".

Proposal

Add the following section:

7.2.9.4 Authorisation

Like a permission, an authorisation is defined by:

-- a behaviour

-- a role or roles involved in that behaviour

--a predicate on behaviour

-- an authority which gives the authorisation

If a set of enterprise objects has this authorisation, then, when the predicate is true, that set of enterprise objects can not be prevented to take part in the action when filling the roles, by order of the authority.

JP15 TL Page 20, 7.9.2 lines 32-33 Obligations, permissions, and prohibitions

Rationale

Since “policy” is extended, “authorization” should also be described under 7.9.2.

Proposal

Add “authorization” sub-clause.

Add the following before like 19.

7.9.2.3 Authorization

An authorization is defined by:

-- a behaviour

-- a role or roles involved in that behaviour

-- a predicate on behaviour

-- an authority that provides authorization

If a set of enterprise objects has this authorization, then, when the predicate is true, that set of enterprise objects is authorized to take part in the action when filling the roles, by order of authority.

USA 86 TH Pages 19-20, 7.9.2.1 to .3, lines 36-39 and 1-33 Obligations, permissions and prohibitions

Rationale

This text prescribes a theory of policy on which there is not consensus. Subclauses 7.9.2.1 thru 7.9.2.3 prescribe a particular language for specifying rules.

It is not in the best interests of the RM-ODP to prescribe in this standard a particular theory of policy or the elements of a language for specifying policy. There is much research in this area and many languages may be suitable. Prescribing a specific language prevents other languages from being ODP compliant.

Proposal

Delete subclauses 7.9.2.1 thru 7.9.2.3 and the Editor's Notes.

7.9.3 Nesting of policy frameworks

The enterprise specification must indicate the possible nesting of communities and thus all the policy rules governing the behaviour of the enterprise objects fulfilling roles in a community. An enterprise object must conform to all policies in all communities it participates.

The enterprise specification must also indicate whether possible dynamic changes in policy rules of the community (for instance, caused by federation negotiations) should be enforced on subcommunities or peer

communities. A community may require that the peer communities co-operating with it follow the same method of adopting dynamic changes to policies as it itself follows.

Note: In practice, this currently means static policies in each community, but a c-object may inherit its policy requirement from several outer communities.

USA 87 TH Pages 20-21, 7.9.3, lines 35-39, 1-4 Nesting of policy frameworks

Rationale

The term "nesting" is unclear; specifically it is unclear whether it implies hierarchy. In addition, policies of various interrelated communities may be mutually inconsistent, and composition rules should prescribe how to determine and resolve such inconsistencies. Thus, the text appears to be overly restrictive. The text in the second paragraph also proposes a way to make specification fragments mutually inconsistent; in fact, composition rules should prescribe the way(s) to resolve these inconsistencies. Finally, the concept of a c-object inheriting its policy requirement is unclear. Thus, the text does not add value; the concept of using composition rules to determine and resolve inconsistencies has been described elsewhere.

Proposal

Remove: The enterprise specification must indicate the possible nesting of communities and thus all the policy rules governing the behaviour of the enterprise objects fulfilling roles in a community. An enterprise object must conform to all policies in all communities it participates.

The enterprise specification must also indicate whether possible dynamic changes in policy rules of the community (for instance, caused by federation negotiations) should be enforced on subcommunities or peer communities. A community may require that the peer communities co-operating with it follow the same method of adopting dynamic changes to policies as it itself follows.

Note: In practice, this currently means static policies in each community, but a c-object may inherit its policy requirement from several outer communities.

Alternatively, rewrite the text along the lines of the Rationale.

7.9.4 Policy violations

TEMPORARY NOTE – The working group considers that text in this subclause should be rewritten using the concept, violation and invites National Body contributions of appropriate text.

FIN66 TL 7.9.4 Policy violations

Rationale

See FIN29

Proposal

Add to the beginning of 7.9.4:

There are violations against explicitly stated policies, i.e., failures (violations against a contract). These should be considered as erroreousness behaviour implementation or specification. There are also violations caused by inconsistent assumptions between communicating parties about the permissions, obligations and prohibitions. These may arise for example in federation situations where there is no full control to the interacting objects environment or in other situations where an action is not considered to be essential enough to be specified with policies in detail for all possible participants of an interaction. Only some of these situations can be considered as design or

implementation errors. Mechanisms are needed for catching violations and for using appropriate recovery or sanction mechanisms.

Policies can be specified as policed and enforced, or unpoliced.

If policies are specified as policed and enforced this can be specified to be by optimistic or pessimistic means.

Pessimistic enforcement is preventative and requires the specification of mechanisms to ensure that the right things are done and the wrong things are not done.

Pessimistic enforcement is specified when trust is low (i.e. when non-compliance is expected to be rife) and the damage caused by non-compliance is potentially high, and when viable preventative mechanisms can be created and/or effective sanctions can be applied post-hoc when non-compliance occurs.

Optimistic enforcement is not preventative. It requires the specification of mechanisms to detect and report/correct non-compliance.

Optimistic enforcement is specified when trust is high and the potential damage due to non-compliance is low, and when viable preventative mechanisms do not exist.

7.9.5 Organisation of policy

FIN67 TH 7.9.5 organization of policy

Rationale

Use of the policy framework for organizing and structuring policy rules.

The concepts of policy type, policy template and policy instance are not a useful classification for this clause. Policies are instances by definition, and they are organised by a policy framework. The implementation of the policy framework may cause restrictions for policy types and even to policy templates when entering actual policies is discussed. However, those discussions are not of suitable abstraction level here.

Proposal

Replacement text:

An enterprise specification and a community specification include a policy framework that defines what kind of policy rules are needed to guide the behaviour of enterprise objects playing various roles in relation to the ODP system in question. The policy rules form a hierarchy and controlled domain in the manner determined by the enterprise specification structure.

Policy rule covers for example

- limits on the system, community or object behaviour in terms of possible interaction sequences with other corresponding entities;
- limits on the quality of service obtained from the peer system, community or object;
- limits on the identity of peer systems, communities, or objects interacting with an entity;
- limits on the technology by which such interactions can be performed;

TEMPORARY NOTE – The working group invites National Body contributions of an explicit description of policy structure at an abstraction level possible and desirable for this purpose. Specifically, this description ought to state that a policy may apply to (collections of) enterprise objects, to (collections of) enterprise objects in roles, to collections of

roles, or to collections of interactions, or to combinations of the above. It is also essential to distinguish between policy types, templates and instances in this context.

The specification of policy covers

-- limits on the community behaviour arising from the permission, or otherwise, of the enterprise objects to take part in interactions; this can be expressed by statements that the various enterprise objects have the capability to perform particular sets of actions;

-- the degree to which, and the circumstances in which, one enterprise object can take over the role and responsibilities of another, acting as a substitute for it.

An enterprise specification may require that successful performance of an interaction between enterprise objects of a community requires that a set of permissions exists. If an enterprise specification requires permissions they are either:

-- associated with a particular target role in that interaction, or

-- associated with that interaction as a whole.

Editor's Note – The meaning of 'target role' may not be clear.

When an enterprise specification requires permissions for an interaction, if permission for that interaction is missing, the interaction fails and therefore the enterprise object may fail to fulfill its role.

Objects can pass permissions between themselves. This passing is itself an interaction, and is subject to same permission rules.

USA 88 TH Page 21, 7.9.5, lines 39-41 Organization of policy

Rationale

This text is not sufficiently clear and precise.

[What it means to pass the absence of a prohibition (i.e., a permission) is not clear to the USA.]

Proposal

Remove:

Objects can pass permissions between themselves. This passing is itself an interaction, and is subject to same permission rules.

Editor's Note – Does this mean subject to the permission rules of this subclause?

7.10 Force rules

FIN68 E7.10 force rules

Rationale

Looks more like definitions than rules for the usage. Only 7.10.3 and 7.10.4 gives further information than the definitions. Questions to ask here: What is the specifier required to say? What would cause contradictory rules? What are the implications of the specification for the community and its environment?

How policies and force rules are related?

An enterprise specification contains policies and rules for changing these policies. The policy changing rules need to specify who are allowed to change the policies. Force rules can be used to express how the chain from parties to the policy changing agents is formed and modified, and how

the responsibilities of actions of the system gets distributed to the parties according to this chain. Thus the access rights and responsibilities become inseparable.

Proposal

JP16 TL Page 22, 7.10 lines 7-8, 18, 20, 26, 27, 36 Force rule

Rationale

It is not clear whether all the parties considered in this clause reside in environment of an ODP system or partially within an ODP system. Assuming all are in the environment, rewrite descriptions without using "machine" concept. See JP8.

Proposal

Reword rules which include "machine" concept in this clause.

Replace with the following texts.

Line 6-7: However, the specification will identify the acts of parties that an ODP system is prepared to participate in, respond to or record.

Line 7-8: The specification will also identify the acts that an agent of a party is prepared to participate in.

Line 18-19: An enterprise specification may also specify delegation from any party to another party or a whole or parts of an ODP system.

Line 19-20: Likewise, an enterprise specification may specify delegation, insofar as provided in delegation by a party, by an ODP system to a party or other ODP system.

Line 26-28: When an ODP system is an agent with authority to delegate, and it may (subject to the provision of that authority) delegate (authority, responsibility, a function, …) to one of its parts, to an ODP system not one of its parts, or to a party.

Line 36-37: -- to further delegate an authority to a party or an ODP system (or, if the agent is an ODP system, to part of the ODP system); this causes the agent delegated to have the authority.

The environment of an ODP system includes parties. Parties can have intentions and are responsible for their actions.

The concepts of subclause 6.4 are used to model an action of a party or parties to which an intention is ascribed. In particular, an act specifies behaviour suited, according to some criteria, to an objective.

Acts of parties are not predetermined by the specification of an ODP system. However, the specification will identify the acts of parties that a machine is prepared to participate in, respond to or record. The specification will also identify the acts that a machine is prepared to participate in as an agent of a party. An enterprise specification describes the authority delegated to an ODP system in terms of:

-- the parties that have delegated authority to the system;

JP16a

Line 6-7: However, the specification will identify the acts of parties that an ODP system is prepared to participate in, respond to or record.

Line 7-8: The specification will also identify the acts that an agent of a party is prepared to participate in.

-- the authority that each party has delegated;

-- the duration and conditions of the delegation;

-- provisions for additional delegation and withdrawal of delegation during the operation of the system.

By each such delegation, that ODP system becomes an agent of the parties delegating, and the parties (collectively) become principal of the system. A principal is responsible for the act of an object acting as its agent.

An enterprise specification may also specify delegation from any party to another party or machine (including a whole ODP system or one of its parts). Likewise, an enterprise specification may specify delegation, insofar as provided in delegation by a party, by a machine to a party or other machine.

JP16b

Line 18-19: An enterprise specification may also specify delegation from any party to another party or a whole or parts of an ODP system.

Line 19-20: Likewise, an enterprise specification may specify delegation, insofar as provided in delegation by a party, by an ODP system to a party or other ODP system.

Each action of an ODP system pursuant to a delegation is an action selected to suit the objectives of a principal, and is therefore an act.

An enterprise specification specifies the force of each act.

7.10.1 Delegation rules

A principal is responsible for the act of an object acting as its agent.

When a machine is an agent with authority to delegate, and it may (subject to the provisions of that authority) delegate (authority, responsibility, a function, …) to one of its parts, to a machine not one of its parts or to a party. That object is then an agent of the principal and has the same authority as if delegated directly by the principal.

JP16c

Line 26-28: When an ODP system is an agent with authority to delegate, and it may (subject to the provision of that authority) delegate (authority, responsibility, a function, …) to one of its parts, to an ODP system not one of its parts, or to a party.

UK26 Cat TH Clause 7.10, page 22, lines 1-29 Machine and delegation

Rationale:

a) Taking into account UK8, “machine” should be replaced by “enterprise object” or “ODP system”, whichever is appropriate.

b) Page 22, line 6. There is a need to clarify: "Acts of parties are not predetermined by the specification of an ODP system.":

i) this relates only to execution of acts that are identified in the specification, since a specification has nothing to say about other acts;

ii) the reference should be to the enterprise specification (of an ODP system).

c) Page 22, line 26. Delete "and" in "and it may" - so that the phrase reads: "When a machine is an agent with authority to delegate, it may etc."

d) Page 22, line 26-29. A whole ODP system cannot delegate responsibility to one of its parts since this mixes levels of abstraction. In the composition corresponding to the ODP system one part must be specified to have the responsibility, which it may delegate according to the delegation rules.

e) There is significant duplication between page 22, lines 8-20 and page 22, lines 26-29. These should be merged into 7.10.1, Delegation rules.

Proposal:

a) Page 22, lines 4-8. Make the following changes:

The execution of ~~A~~acts of parties is~~are~~ not predetermined by an enterprise~~the~~ specification ~~of an ODP system~~. However, the enterprise specification identifies~~will identify~~ the acts of parties that an ODP system~~a machine~~ is prepared to participate in, respond to or record. The enterprise specification ~~will~~ also identifies~~identify~~ the acts that any enterprise object that is not a party~~a machine~~ is prepared to participate in as an agent of a party.

b) Move page 22, lines 8-20 to 7.10.1and merge with page 22, lines 26-29 and replace page 22, lines 25-29 by:

An enterprise specification may specify delegation from any party to another party or to an enterprise object that is not a party~~machine (including a whole ODP system or one of its parts)~~.An enterprise specification describes the authority delegated to an enterprise object~~ODP system~~ in terms of:

-- the parties that have delegated authority to the system;

-- the authority that each party has delegated;

-- the duration and conditions of the delegation;

-- provisions for additional delegation and withdrawal of delegation during the operation of the system.

By each such delegation, that enterprise object~~ODP system~~ becomes an agent of the parties delegating, and the parties (collectively) become principal of the enterprise object~~system~~. A principal is responsible for the acts of an enterprise object acting as its agent.

Likewise, an enterprise specification may specify delegation, insofar as provided in delegation by a party, by an enterprise object that is not a party to another enterprise object, whether or not that enterprise object is a party~~a machine to a party or other machine~~. That enterprise object is then an agent of the principal and has the same authority as if delegated directly by the principal.

7.10.2 Authority rules

For each authority delegated, an enterprise specification describes the force of acts of an agent in exercising that authority. The authority delegated may be:

-- to make a commitment; this binds the principal

-- to issue a declaration; this establishes the truth of some proposition just as if the principal had made the declaration;

-- to further delegate an authority to a party or machine (or, if the agent is a machine, to part of the agent); this causes the agent delegated to have the authority;

JP16d

Line 36-37: -- to further delegate an authority to a party or an ODP system (or, if the agent is an ODP system, to part of the ODP system); this causes the agent delegated to have the authority.

UK27 Cat E Clause 7.10, page 22, line 1 to page 23, line 20 Force rules

Rationale:

Page 22, Lines 36-37. This bullet should be the last bullet in the list and should be simplified to "to further delegate an authority; this causes the agent delegated to have the authority" (since delegation rules are in 7.10.1).

Proposal:

Page 22, Lines 36-37. Make this bullet the last bullet in the list and change to:

-- to further delegate an authority; this causes the agent delegated to have the authority;

-- to make a prescription that establishes a rule; such a rule has the same force as if the principal had made the prescription.

7.10.3 Commitment rules

An enterprise specification identifies, for every commitment, the obligation created. It identifies, for every commitment made by an agent, the principal(s) obligated.

Establishing behaviour in an enterprise specification includes commitments by the objects participating in the establishing behaviour. If the establishing behaviour is implicit, it includes prescriptions that apply to the objects in the resulting liaison.

7.10.4 Declaration rules

A declaration identifies the changes that take place in the environment of an object as the result of an internal action of that object. An enterprise specification defines the conditions required for a particular declaration to be effective.

NOTE – A declaration may not be effective (cause the change in the environment of the object) until some interaction of the object such as, for example, a publication.

7.10.5 Prescription rules

An act of an enterprise object will be a prescription only when:

-- that object is a party that by its nature may establish rules,

-- that object is an agent of such a party, delegated authority to establish rules on behalf of that party,

-- the specification explicitly provides for those actions of that object that will be prescriptions, or

-- that object is, in a previous epoch, specified to establish rules.

An important special case of delegation is where the authorized action is a prescription; that is, when the delegation enables an enterprise object to make a prescription.

8 Consistency rules

FIN69 E 8 clause title

Rationale

Readability.

Proposal

Remove subtitle 8.1. and use the title Viewpoint correspondences for clause 8.

UK28 Cat TH Clause 8 to 11 Ordering of clauses

Rationale:

These clauses are not in the order agreed at the Madrid meeting.

Proposal:

a) Move current Clause 9 Relations Between Standards and Product Development, to be Clause 8;

b) Move current Clause 10 Enterprise Language Compliance, to be Clause 9;

c) Move current Clause 11 Conformance and reference points, to be Clause 10;

d) Move current Clause 8 Consistency rules, to be Clause 11.

8.1 Viewpoint correspondences

FIN6 G Summary of concerns Viewpoint correspondencies

Proposal

See FIN70-FIN76

The underlying rationale in identifying correspondences between different viewpoint specifications of the same ODP system is that there are some entities that are represented in an enterprise viewpoint specification, which are also represented in another viewpoint specification. The requirement for consistency between viewpoint specifications is driven by, and only by, the fact that what is specified in one viewpoint specification about an entity needs to be consistent with what is said about the same entity in any other viewpoint specification. This includes the consistency of that entity's properties, structure and behaviour.

JP17 E Page 24, 8.1 line 2 Viewpoint correspondence

Rationale

Typo

Proposal

Change the tense.

Replace "related" by "relates."

USA 89 TH Page 23, 8.1, lines 23-28 Viewpoint correspondences

Rationale

The text refers to entities which exist in UoD. Objects are models of entities. Thus, if the term "entities" is used then entities are "modelled" rather than "represented" in a viewpoint specification. In addition, the text refers only to entities in isolation, and thus does not directly deal with the need for collective state and behaviour specified in one viewpoint specification to be consistent with collective state and behaviour specified in another viewpoint specification. A better terminology can be borrowed from the definition of a proposition, 2-6.2.

Proposal

Change: "Represented"

To:

"modelled"

Change: ...about an entity... ...about the same entity...

To:

...about one or more entities... ...about the same entities...

The specifications produced in different ODP viewpoints are each complete statements in their respective languages, with their own locally significant names, and so cannot be related without additional information in the form of correspondence statements. What is needed is a set of statements that make clear how constraints from different viewpoints apply to particular elements of a single system to determine its over all behaviour. The correspondence statements are statements that relate the various different viewpoint specifications, but do not form part of any one of the five basic viewpoints. The correspondences can be established in two ways:

-- by declaring correspondences between terms in two different viewpoint languages, stating how their semantics interact. This implies that the two languages are expressed in such a way that they have a common, or at least a related, set of foundation concepts and structuring rules. Such correspondences between languages necessarily imply and entail correspondences relating to all things of interest which the languages are used to model (e.g. things by objects or actions);

-- by considering the extension of terms in each language, and asserting that particular entities being modelled in the two specifications are in fact the same entity. This related the specifications by identifying which observations need to be interpretable in both specifications.

There are two kinds of standardization requirements relating to correspondences:

-- Some correspondences are required in all ODP specifications; these are called required correspondences. If the correspondence is not valid in all instances in which the concepts related occur, the specification simply is not a valid ODP specification.

-- In other cases, there is a requirement that the specifier provides a list of items in two specifications that correspond, but the content of this list is the result of a design choice; these are called required correspondence statements.

FIN 10b

Proposed text for clause 8, add between lines 10 and 11 at page 24:

When this standard is taken as a basis for a standard methodology or a tool supporting a detailed software engineering methodology, the tool or methodology designer can specify a list of correspondences between viewpoint concepts or between enterprise language and the modeled system concepts.

FIN70 TL 8, lines 10-11 categorization expansion

(part of FIN10)

Rationale

The situations of interest

-level of requirements stated in RM ODP

-requirements and interpretations build in into tools and methods

-relationships stated in an individual specification

Proposal

Proposed text for clause 8, add between lines 10 and 11 at page 24:

When this standard is taken as a basis for a standard methodology or a tool supporting a detailed software engineering methodology, the tool or methodology designer can specify a list of correspondences between viewpoint concepts or between enterprise language and the modeled system concepts.

The minimum requirement for consistency in a set of specifications for an ODP system is that they should exhibit the correspondences defined in the Reference Model (part 3 clause 10), those defined in this standard, and those defined within the specification itself.

NOTE – The following clauses identify the correspondences between the enterprise viewpoint and the information, compuational and engineering viewpoints. Although in particular models it may be possible to establish correspondences between instances of enterprise concepts and instances of technology concepts, there are no useful generic correspondences of this nature. In particular, it should be noted that although 'enterprise wide' policies may exist about adoption of particular technologies, such statements are not enterprise issues as such, and should therefore appear in the technology specification for the system. Only in cases where the system has some behaviour that is related to such technology policy (for example if the system was concerned with the management of procurement of IT systems), would such policy appear in the enterprise viewpoint specification.

8.2 Enterprise and information specification correspondences

8.2.1 Concepts related by correspondences

The enterprise concepts related are:

-- Objective;

-- Community;

-- Enterprise object;

-- Scope;

-- Role;

-- Actor;

-- Artefact;

-- Process;

-- Policy.

FIN71 TH 8.2.1 correspondences / EL and info

Proposal

Delete bullets actor and artefact; replace process by action.

JP18 E Page 24, 8.2.1 lines 24- Concepts related by correspondence

Rationale

Consistency between 8.2.1 and 8.2.3 is required.

Proposal

Add more enterprise concepts which may have correspondence with information specification, or remove some of them so that only concepts listed appears in clause 8.2.3 Required correspondence statements.

Add the following concepts.

-- Step

-- Resource

Or, remove the following.

-- Objective

-- Scope

-- Actor

-- Artefact

-- Process

The information concepts related are:

-- Information object;

-- Dynamic schema;

-- Static schema;

-- Invariant schema.

8.2.2 Required correspondences

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

There are no required correspondences.

8.2.3 Required correspondence statements

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

The specifier shall provide:

-- for each enterprise object in the enterprise specification, a list of those information objects that describe part or all of its state;

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for n:m correspondences.

FIN72 TH 8.2.3 correspondences /EL and info

Proposal

a) Replace temporary not on lines 7-8 with

Note – An information object may appear in lists of multiple enterprise objects.

…

ES28 E page 24, 8.2.3, lines 31-32 Temp. note on n:m correspondences

Rationale

We think that there may be no need for n:m correspondences, since an enterprise object can be a composite object, so 1:m correspondences may cover all cases.

Proposal

Include a Note stating this fact.

-- for each role in each community in the enterprise specification, a list of those information object types that describe part or all of the state associated with the roles it fills;

FIN72b TH 8.2.3 correspondences /EL and info

Proposal

…

b) Replace “with the roles it fills” on line 10 with “filling this role”.

-- for each policy in the enterprise specification, a list of the invariant, static and dynamic schemata of information objects modified by the policy; an information object is included if it is associated with the enterprise community that is subject to that policy;

-- for each action in the enterprise specification, the information objects that have a dynamic schema constraining that action;

FIN72c TH 8.2.3 correspondences /EL and info

Proposal

…

c) Replace bullet on lines 14-15 with

- for each action in the enterprise specification, the information objects that have a dynamic schema further contraining the realization fo that action

as the action does not become defined by these contstraints.

TEMPORARY NOTE - The working group invites National Body comment and contributions on whether this constraint is necessarily a definition of the action.

-- for each relationship between enterprise objects, the invariant schema that represents it.

8.3 Enterprise and computational specification correspondences

8.3.1 Concepts related by correspondences

The enterprise concepts related are:

-- Enterprise object;

-- Actor role;

-- Enterprise interaction;

FIN73 TH 8.3.1 correspondences / EL and computational

Proposal

a) Remove actor role and on line 23.

b) Add on line 23

Note: actor and interface roles are related to enterprise objects within the enterprise specification.

c) Replace line 24 “enterprise interaction” with

- interaction between roles

Note – action between roles are essentially captured by interactions.

-- Policy.

JP19 E Page 25, 8.3.1 line 23-24 Concepts related by correspondence

Rationale

Undefined term should not be used.

Proposal

Give definition or remove a bullet.

Replace line 23 by the following.

-- Actor

Remove line 24.

The computational concepts related are:

-- Computational object behaviour;

-- Computational interaction;

-- Computational interface;

-- Binding object.

8.3.2 Required correspondences

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

There are no required correspondences.

8.3.3 Required correspondence statements

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

The specifier shall provide:

FIN74 TH 8.3.3 correspondences / EL and computational

Proposal

a) Add in the beginning of the list

- for each enterprise object, a configuration of computational objects for realizing the required behaviour

Note – the identification and introduction of enterprise objects fulfilling an actor role can be indirect, basaed on elaborate assignment rules and supporting system functionality

Note – the configuration can be expressed either direcly or indirectly as a set of rules for constructing the configuration.

-- for each interaction in the enterprise specification, a list of those computational interfaces and operations or streams that correspond to the enterprise action, together with a statement of whether this correspondence applies to all occurrences of the action, or is qualified by a predicate;

FIN74b TH 8.3.3 correspondences / EL and computational

b) Page 26, lines 1-2 remove “or is qualified by a predicate” and add note

Note – if the same operation or stream can be used for multiple enterprise viewpoint actions, the computational specification has to include predicates for qualifying each case.

-- for each role affected by a policy in the enterprise specification, a list of the computational object types that exhibit choices in the computational behaviour that are modified by the policy;

-- for each interaction between roles in the enterprise specification, a list of computational binding types that are constrained by the enterprise interaction;

FIN74c TH 8.3.3 correspondences / EL and computational

c) Page 26, line 6, replace “that are constrained by the enterprise interaction” by “required by the enterprise interaction or a rule for identifying such bonding types”.

-- for each enterprise interaction type, a list of computational behaviour types capable of representing (i.e. acting as a carrier for) the enterprise interaction type.

FIN74d TH 8.3.3 correspondences / EL and computational

d) Page 26, line8, add to the end “or a rule for identifying such behaviour type.”

8.4 Enterprise and engineering specification correspondences

8.4.1 Concepts related by correspondences

The enterprise concepts related are:

-- Enterprise object;

-- Principal role;

-- Actor role;

-- Policy

The engineering concepts related are:

-- Basic engineering object;

-- Node;

-- Stub;

-- Binder;

-- Protocol object;

-- Interceptor.

FIN75 TH 8.4.1 correspondences / EL and engineering

Proposal

Remove bullets principal role, actor role and basic engineering object.

JP20 E Page 26, 8.4.1 lines 13-15

Rationale

Consistency between 8.4.1 and 8.4.3 is required.

Proposal

Unused terms should not be part of 8.4.1.

Remove line 13 through 15.

Remove line 17 and line 20.

Add the followings.

-- Channel

8.4.2 Required correspondences

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

There are no required correspondences.

8.4.3 Required correspondence statements

TEMPORARY NOTE - The working group invites National Body comment and contributions on the need for additional correspondence rules in this sub-clause.

The specifier shall provide:

-- for each enterprise object in the enterprise specification, a list of those engineering nodes that support some or all of its behaviour;

-- for each interaction between roles in the enterprise specification, a list of engineering channel types and stub, binding or protocol objects that are constrained by the enterprise interaction;

FIN76 TH 8.4.3 correspondences / EL and engineering
Proposal
a) Replace bullets on lines 31-34 by

- for each enterprise object in the enterprise specification, rules about assigning some or all of its behaviour to engineering nodes; these rules may capture policies from enterprise specification;

- for each interaction between roles in the enterprise specification, a list of engineering channel types and stub, binder or protocol objects, and interceptors that are required by the enterprise interaction; these rules are constrained by enterprise policies.

9 Relations between standards and product development

FIN80 E8 and 9 readability
Proposal
Swap clauses 8 and 9.

TEMPORARY NOTE - The working group invites National Body to propose another title for this clause.

FIN77 TL 9, 9.1 Title, scope of discussion
Proposal
Remove subtitle 9.1 and use the title “Compliance, conformance and consistency” for clause 9.

9.1 Compliance, conformance, testing and consistency

This standard follows the normal practice in standardization work in using the term compliance to describe the relationship between two standards. One standard complies with another if it makes correct use of the ideas, vocabulary or framework defined there. This implies that, if a specification is compliant, directly or indirectly, with some other specifications, then the propositions which are true in those specifications are also true in a conformant implementation of the specification.

The term conformance is used for the relationship between some product or artefact and the specification from which it is produced. Conformance can be tested by inspecting the product produced to confirm the claim that its properties or behaviour are as required by the standard.

In ODP specifications, there is a need for the specifier to declare those points at which tests are to be performed and for the implementor to identify those points when offering the product for test. Large specifications are frequently organized into a specification framework populated by more detailed component specifications. The framework identifies a wide range of points at which observations can, in principle be made. These points are called reference points. The subset of reference points where tests of an implementation are required by the more detailed specifications are called the conformance points for that specification.

ODP systems are specified in terms of a number of viewpoints, and this gives rise to an accompanying requirement for consistency between the different viewpoint specifications. The key to consistency is the idea of correspondences between specifications; i.e., a statement that some terms or structures in one specification correspond to other terms and structures in a second specification. Correspondences can be identified between

two different viewpoint specifications in a single language or in two different languages. Statements of correspondences between two languages imply equivalent correspondences between any pair of specifications expressed in those languages.

JP21 TL Page 27, 9.1 lines 20-23 Compliance, conformance, testing and consistency

Rationale

RM-ODP Part 3 Clause 10 includes discussion about correspondence between two different specifications in a single language, which seems to imply two different (enterprise, information, computational, engineering, technology) specifications described in the same viewpoint language. However, the sentence starting middle of line 20 seems to use "language" in different sense.

Proposal

Clarify the meaning of "language" and reword the text appropriately.

9.2 Completeness

Specifications can be produced as a prelude to implementation, and generally change during implementation or to support system evolution. Specifications can also be produced to capture the properties of existing systems or components in order to facilitate their reuse. The references to the process of specification in this clause are intended to cover both these situations.

When a set of viewpoint specifications and correspondences is created for an ODP system, a succession of design choices is made, gradually reducing the number of conceivable implementations that would be consistent with the specification. This process is never absolutely complete, since there are always implementation choices and changes in circumstances in the environment that affect the system's behaviour, but there is some point in the design process when the specifier judges that the specification is sufficiently complete to reflect their purpose. At this point, the specification is said to have reached the viable stage. This is the stage in the specification process where it would be possible to produce some worthwhile implementation. This statement does not imply that the specification is, in any way, frozen.

The viable stage depends on the purpose of the specification, because there may be significant differences in the degree of completeness expected in, for example, an accounting policy applied to a range of independent machines or to an inter-organizational workflow. The viable stage will not be assessed to be the same for all possible applications of the of any particular specification notation.

FIN78 E9.2, page 27, line 40 Spelling

Proposal

Remove words "of the".

9.3 Scoping statement

FIN17b TH 6.1.x, 9.3 Scoping statement

Rationale

See clause 8 and capture that rationale and semantics into a definition.

Proposal

…

Remove 9.3.

FIN79 E9.3 consistency with other comments made

Rationale

See FIN17.

Proposal

Remove clause 9.3.

Any specification should include a scoping statement that specifies the preconditions for the use of that specification. A scoping statement says whether a specification is appropriate in a given situation, and must be satisfied before it makes sense to make observations of the real world and compare these with specified observable properties to test conformance to the specification.

The provision of an accurate scoping statement is particularly important if reuse of the enterprise specification is expected. It allows the specifier who might incorporate the existing specification fragments to ask "is this specification for me?" before they begin to ask "what must my enterprise and its supporting systems do?"

The scoping statement expresses the agreed scope of the system being specified, in terms of what it is intended to do, and what properties the environment must have for the system to operate.

The scope of the system is dependent on choices made by the specifier. Each specification needs to be considered separately for completeness, compliance and conformance, as the specifier finds appropriate under the prevailing circumstances.

10 Enterprise Language Compliance

An enterprise specification compliant with this Recommendation | International Standard shall use the concepts defined in clause 6 and those in clause 5.1 of ITU-T Rec. X.903 | ISO/IEC 10746-3, as well as the concepts defined in ITU-T Rec. X902 | ISO/IEC 10746-2, structured as specified in clause 7.

Concepts from ITU-T Rec. X.902 | ISO/IEC 10746-3 not refined in this Recommendation | International Standard may also be employed. Where such concepts are employed, the specification concerned shall include explanations of the relationships between the concepts concerned and those defined in clause 6.

ES 29 EPage 27, 10, line 37 Typographic error

Rationale

There is a small typo error in line 37, that mentions "ITU-T Rec X.90**2** | ISO-IEC 10746-**3**". It should be "ITU-T Rec X.902 | ISO-IEC 10746-**2**"

Proposal

Change it accordingly.

USA 90 TH Page 28, 10, lines 12-17 Enterprise Language Compliance

Rationale

Concepts defined in Part 2 are to be used in all viewpoint specifications. [3-4.2.2] Furthermore, "objects identified in one viewpoint can be specified using the viewpoint language associated with that viewpoint or using the viewpoint languages associated with other viewpoints." [3-4.2.2]

Proposal

Change:

An enterprise specification compliant with this Recommendation | International Standard shall use the concepts defined in clause 6 and those in clause 5.1 of ITU-T Rec. X.903 | ISO/IEC 10746-3, as well as the concepts defined in ITU-T Rec. X902 | ISO/IEC 10746-2, structured as specified in clause 7.

~~Concepts from ITU-T Rec. X.902 | ISO/IEC 10746-2 not refined in this Recommendation | International Standard may also be employed. Where such concepts are employed, the specification concerned shall include explanations of the relationships between the concepts concerned and those defined in clause 6.~~

Concepts from other modelling languages may also be employed. Where such concepts are employed, the specification concerned shall include or refer to definitions of each such concept, in terms of the concepts defined in clause 6, in ITU-T Rec. X.902 | ISO/IEC 10746-2, or in clause 5.1 of ITU-T Rec. X.903 | ISO/IEC 10746-3 , and explanations of the relationships between such concepts and those defined in clause 6.

11 Conformance and reference points

This standard defines the Enterprise Language, which provides a framework for a variety of notations to be used in specification. As such it creates a formal system that does not itself involve conformance, any more than, say, a programming language grammar involves conformance. However, specific notations derived from this standard will be supported by (generally automated) tools and design processes that produce and maintain enterprise specifications for systems, and the conformance of these tools and processes can be tested. This includes the generation of specifications that conform to the structural or grammatical rules of the language, and the construction of systems which, in operation, perform in a way consistent with the semantics of the language.

In general, such tools and processes manipulate not only the enterprise viewpoint specification but also manage correspondences with other viewpoint specifications, and so wider issues of conformance to complete sets of ODP specifications need to be considered.

NOTE: There are correspondences between each possible pair of viewpoint specifications, but the conformance issues involved are particularly important in this standard because the policies expressed in the enterprise specification are reflected in all the other viewpoints.

The Enterprise Language places requirements on organizational structures and business processes that cannot be observed directly, but must be deduced from the variety of interactions between the system or systems involved and their environment. In claiming conformance to an enterprise specification, the system provider must state what observable reference points in the system are conformance points, and how observations at these points can be interpreted to correspond to enterprise concepts. With this information, a tester of the system is in a position to determine by observation whether the system behaves correctly. In ODP, conformance is based on the declaration of Engineering Viewpoint reference points (in clauses 5-7 of ITU-T Rec. X.903 | ISO/IEC 10746-3), and the implementor of an Enterprise Specification must state correspondences to the engineering viewpoint in order to relate observations at the engineering reference points to enterprise concepts.

Annex A Overall structure of an enterprise specification

FIN81 TH Annex A consistency with the text
Rationale
Diagram needs redrawing. Problems include
-Relationships of objective, state, and role
-Party and owner being on their own
-C-object exporting something
-Scoping statement being not connected to anything
-Relationships of a system restricted to c-object, purpose and scope

This is a part of a meta-model of the enterprise viewpoint. This meta-model defines the structure of an enterprise specification.

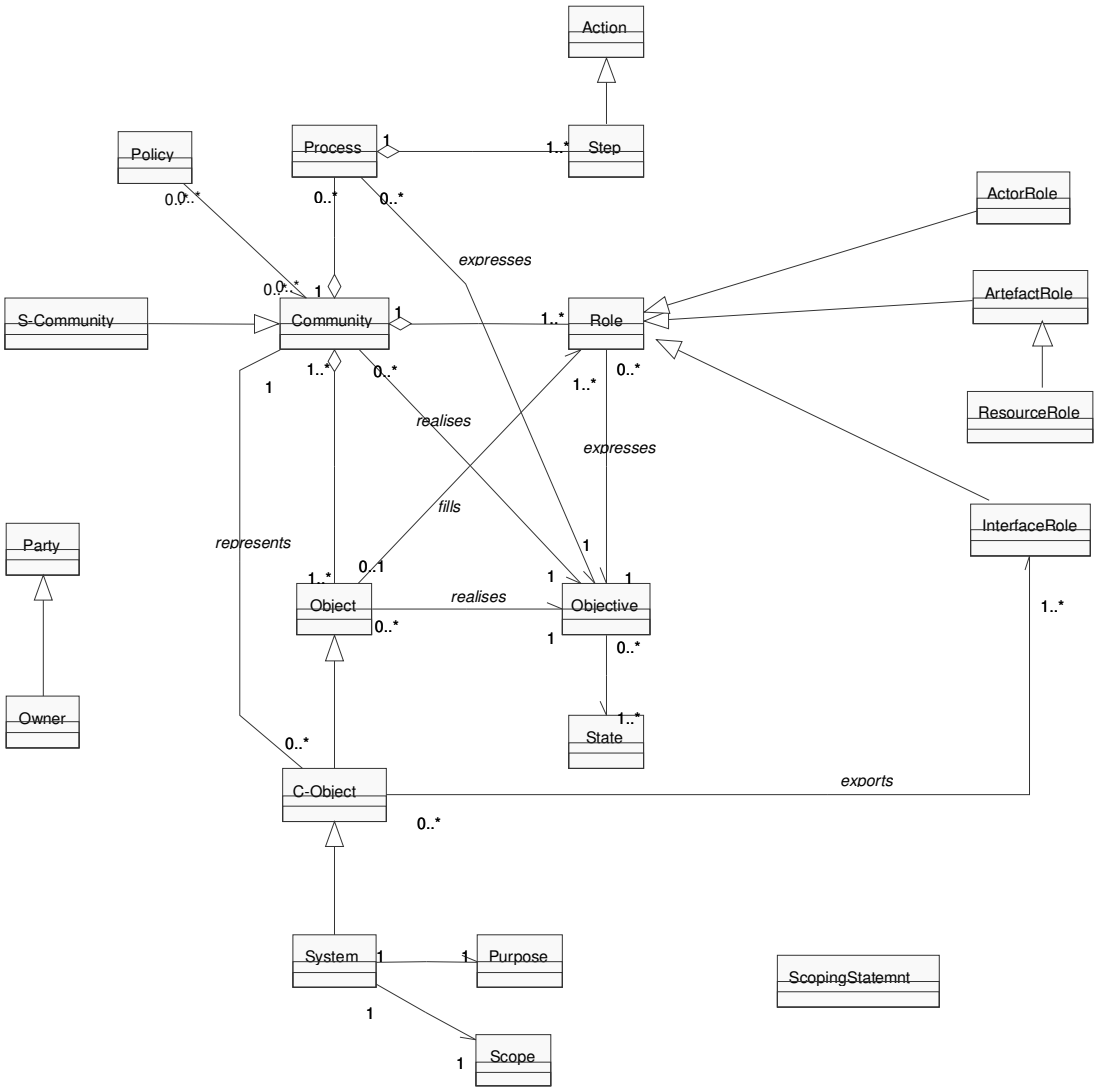
NOTES:

1 – This meta-model is not complete, but it is helpful for the understanding of the enterprise viewpoint.

2 – Some links that come from RM-ODP Part 2 are not defined in this meta-model (i.e. the fact that a role is an identifier for a behaviour, et cetera).

3 – This meta-model is compliant with the ODP Type Repository Function (ISO/IEC FDIS 14769: Information Technology – Open Distributed Processing – Type Repository Function).

4 – The notation used in this meta-model is UML (ISO/IEC DIS 19501-1, Information Technology – Unified Modeling Language (UML) - Part 1: Specification).



AFNOR 24 TH Annexe A
Rationale
As the standard has changed, the meta-model must change
Proposal
This is a new meta-model. This meta-model is MOF compliant.
It represents the concepts of : Template, Type, Action, State, Objective, Behaviour, Object, C-Object, Community, Role, Process, Step and Policy.
Two classes have been added to simplify the meta-model : ModelElement and BehaviouralElement

A Type has an attribute named type which is a String representing the predicate. A Type can be linked to one or more ModelElement.

A Template can be linked to one or more ModelElement.

A ModelElement can be linked to zero or more Type and it can be linked to zero or more Template.

Action, Behaviour, Process and BehaviouralElement inherit from ModelElement.

An Action has an attribute named name which is a String representing the name of the Action. An Action is linked to two States. The State before the Action and the State after the Action. An Action can be linked to zero or more Objects (the objects realizing the action). An Action can be referenced by a Behaviour.

A Behaviour references Actions. It can be identified by a Role. It can be exhibited by a BehaviouralElement. A Behaviour can be a scope.

An Objective prefers one ore more States. An Objective has a an attribute named name which is a String representing the name of the Objective.

A BehaviouralElement has an attribute named name which is a String representing the name of the BehaviouralElement. A BehaviouralElement can have a Behaviour and it can have an Objective. Communities and Objects are BehaviouralElements.

An Object is a BehaviouralElement that has an attribute named kind which can be System, Party, Machine or Resource. An Object can fulfil some Roles and it can performs some Actions.

A Community is a BehaviouralElement that includes Roles and groups Objects. A Community can have some processes. A Community can be an S-Community.

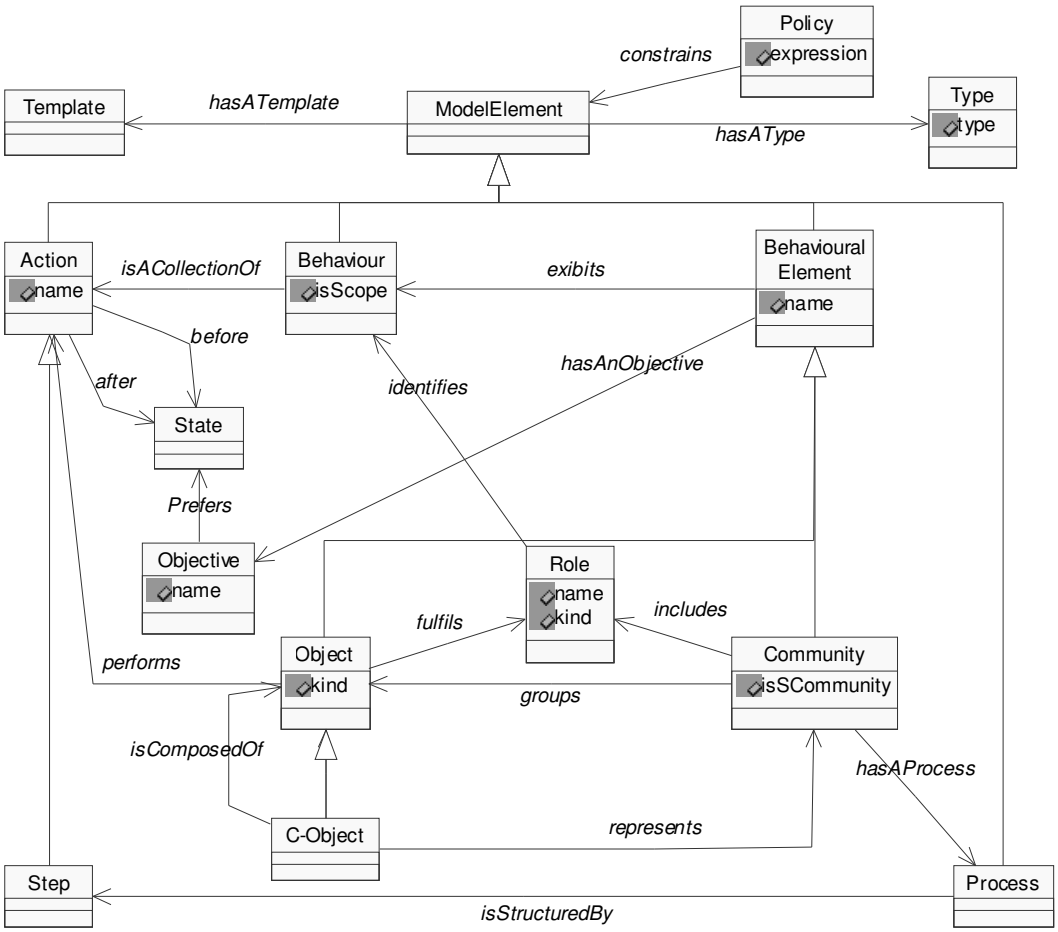
A C-Object is an object that is composed of several Object and that represents a Community.

A Role has an attribute named kind which can be Artefact, Actor, Resource or Interface. A Role is included in a Community. A Role may be fulfilled by an Object. A Role identifies a Behaviour.

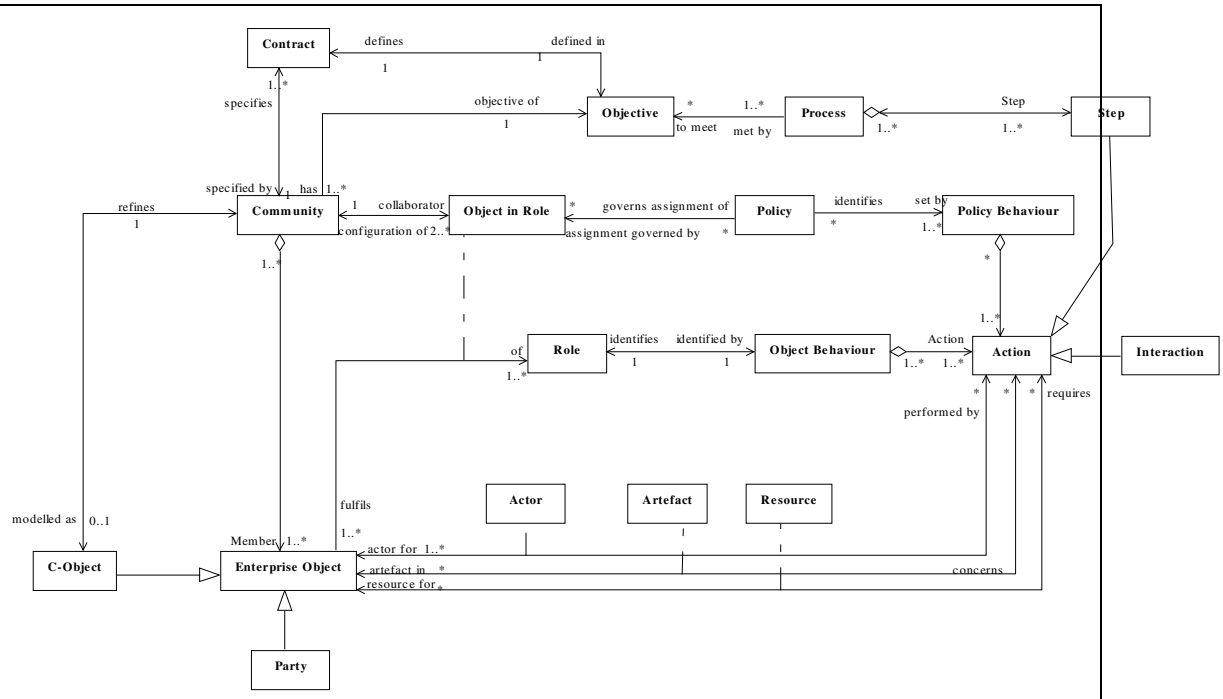
A Process is structured by some Steps.

A Step is an Action.

A Policy has an attribute named expression which is a String representing the set of rules.



UK29 Cat TL Annex A	Overall structure of an enterprise specification
Rationale:	
The annex comprises a diagram and short text proposed by AFNOR. It requires review for completeness.	
Proposal:	
Replace the diagram with the diagram below	



USA 91 TH Pages 30-31 Annex A

Rationale

The Annex as presented now has very weak semantics and appears not to correspond to the normative text. (As a simple but important example, “A step in a process may itself be modelled as a (sub-)process.” is not shown.) If UML is to be used for representation purposes then all UML elements chosen ought to have well-defined semantics. It is possible to do that for relationships provided that the semantics of the relationships will be defined using UML stereotypes. These relationships will include RM-ODP composition as defined in 2-9.1 and RM-ODP subtyping as defined in 2-9.9. These relationships do not have to be binary; if their semantics (invariants) so requires they will not be binary. For all other relationships, their invariants (rather than just names) should be provided. Some relationships are not shown at all, such as between an owner and a community (and other <X>s); neither is an <X> shown. Finally, arrows at the end of some relationships should not be shown.

Proposal

Redraw the diagram in accordance with the Rationale after the normative text will be finished.

Index

<X> domain, **4**
<X> federation, **5**
<X> template, **4**
act, **9**
actioj, 9
action, **4**, 7, 8, 9
activity, 7
activity structure, 7
actor, **8**
actor role, **8**
agent, **8**, 9, 22
artefact, **8**
artefact role, **8**
authorisation, 8, 9
authority, 8
behaviour, 7
behaviour, 8, 9, 18
c-object, **7**
commitment, **9**, 22
community, **5**, 8, 18
composite object, **4**
composition, **4**
computational interface, **5**
configuration (of objects), **4**
conformance, **4**
conformance point, **4**
contract, **4**, 9
contracting party, **9**
declaration, **9**, 22
delegate, 8
delegation, 8, **9**
dynamic schema, **5**
entity, **4**, 7
environment (of an object), **4**
environment contract, **4**
epoch, **4**
establishing behaviour, 22
incremental modification, **4**
instance (of a type), **4**
instantiation (of an <X> template), **4**
interface, **4**
interface role, **8**
internal action, **4**
interworking reference point, **4**
invariant, **4**
invariant schema, **5**
liaison, **4**, 22
location in space, **4**
location in time, **4**
machine, 22
object, **4**
objective (of an <X>), **7**
obligation, **4**, 9, 22
ODP standards, **4**
ODP system, **4**
owner (of an <X>), **8**
party, **7**, 8, 9, 22
perceptual reference point, **4**
permission, **4**, 9
policy, **4**, 5, **9**
postcondition, **4**
precondition, **4**
prescription, **9**, 22
principal, **8**, 22
process, **7**
programmatic reference point, **4**
prohibition, **4**, 9
proposition, **4**
purpose (of a system), **7**
purposeful selection, 9

quality of service, **4**
reference point, **4**
refinement, **4**
resource, **8**
resource role, **8**
responding object, **4**
role, **4**, 8
rule, 9, 22
s-community, **7**
scope, **7**
state (of an object), **4**
static schema, **5**
step, 7
subtype, **4**
system, **4**, 7
template, 7
type, 7
type (of an <X>), **4**
valuation, **9**
viewpoint (on a system), **4**
violation, **9**