UK Comments on ISO/IEC JTC1/SC7 N2359 – FCD of IS15414

ISO/IEC JTC1/SC7

Title: FCD 15414: Information Technology - Open Distributed Processing - Reference Model – Enterprise Viewpoint.

Project: 07.77

Vote:

__ APPROVAL OF THE DRAFT AS PRESENTED

__ APPROVAL OF THE DRAFT WITH COMMENTS AS GIVEN ON THE ATTACHED

__ general:

__ technical:

__ editorial:

X DISAPPROVAL OF THE DRAFT FOR REASONS ON THE ATTACHED

__ Acceptance of these reasons and appropriate changes in the text will change our vote to approval

__ ABSTENTION (FOR REASONS BELOW):

P-member voting: UK

Date: 9 January 2001

Introduction

The UK offers the following comments on the Final Committee Draft of IS 15414, Information Technology - Open Distributed Processing - Reference Model – Enterprise Viewpoint., published as ISO/IEC JTC1/SC7 N2359 on 19 July 1999.

This document is formatted in accordance with the SC7 conventions for Ballot Comment and uses the following categorisation of comments.

Category	Description	Impact
G	General	Applicable generally i.e. in multiple places throughout
E	Editorial	Cosmetic including typographical and grammatical
TL	Technical Low (Minor)	Rejection of the comment would not prevent changing a negative vote to a positive one.
TH	Technical High (Major)	Considered to be an immovable position i.e. a negative vote would remain unless this was satisfactorily addressed

General Comments

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.

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.

Comments on Clause 5

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.

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

Comments on Clause 6

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>)

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)

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)

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

[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.

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.

Comments on Clause 7

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.

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".

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)."

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:

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.

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."

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

b) Replace Page 14, line 11 with:

7.5.2 <X>-federation community type

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.

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.

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.

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.~~

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

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.

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?~~

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~~.

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.

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.

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.

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;

Comments on Clauses 8 to 11

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.

Comments on Annex

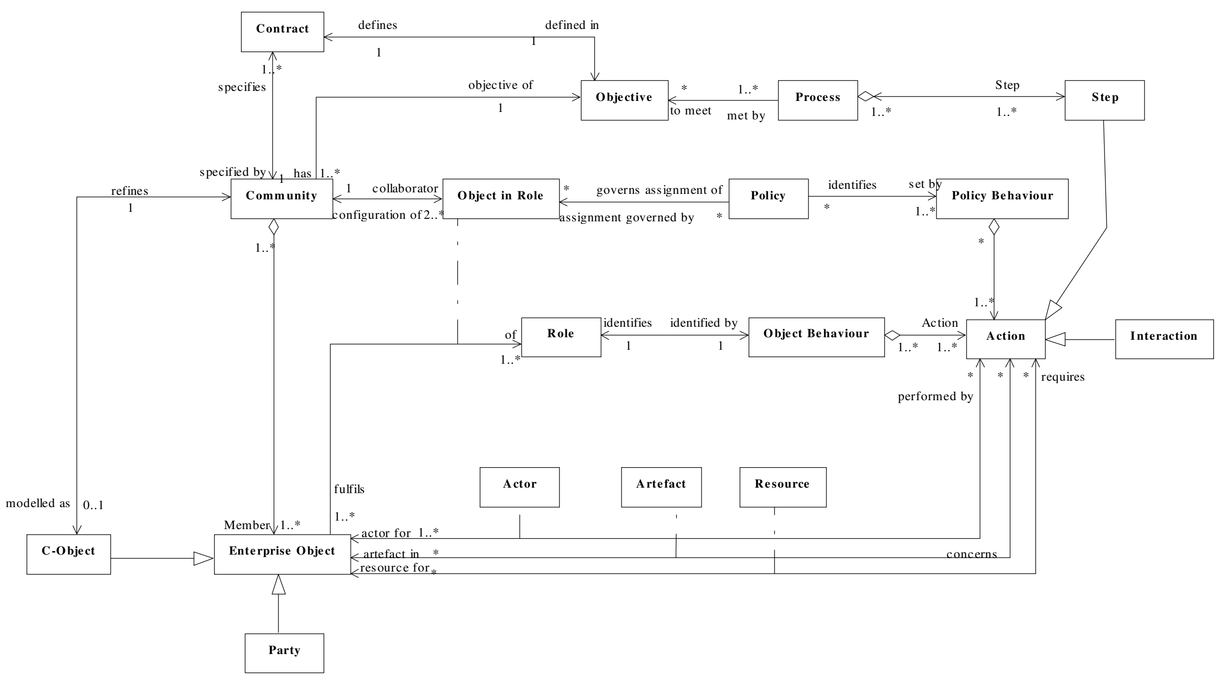
UK29 Cat TL Annex A Overall structure of an enterprise specification

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



Annex to UK Comments on FCD 15414: Information Technology - Open Distributed Processing - Reference Model – Enterprise Viewpoint.

A discussion of action occurrence

[This text is extracted from an unpublished paper on the specification of policy in ODP enterprise specifications]

To understand the relationship between action occurrence and policy, it is necessary to spend some time clarifying the basic ideas associated with specification. Experience in trying to establish the basic concepts for describing ODP systems has shown that different people put subtly different emphasis on some of these concepts, based on their various backgrounds and concerns, and while these can be reconciled once identified, they can cause quite deep-seated misunderstandings if not appreciated. The problem is centered on the concept of occurrence of the distinction between type and instance.

However, first we need to distinguish between occurrence of features in the system following the specification and occurrence in the specification. Within the specification, we can define types as a short-hand for the common properties of other elements - to establish a classification. A specification can define a type - a date-type, say - and then declare a number of objects as being instances of this type. Considering the specification on its own, one can then talk of these objects as being instances of the type, and say that the object "myBirthDay" and the object "yourBirthDay" are different occurrences of the type "date". On the other hand, one may concentrate on the relationship between the specification and some real-world object that implements it, and so is, in a different sense, an occurrence of it. In this sense the specification as a whole can be seen as defining a type, which in the context of the possible execution environments has instances wherever and whenever the specification is applied.

The touchstone for the correspondence between specification and the thing in the real world being specified is observation. A real-world thing corresponds accurately to its specification if and only if some observable set of events in the real-world corresponds to the behaviour required in the specification. However, the word "corresponds" hides a great deal of complexity. It involves an ill-defined degree of interpretation and quantification, and it is here that the problem lies.

Where is the ambiguity? It lies in the scope of any claim of correspondence made. Consider one action within the specification, say the input of a date by a user. The claim may be applied to:

* Some particular historical trace, in which a single observation was made at a certain time and place. This is commonly the case when debugging from a record of execution. The action occurs exactly once, and is associated one-to-one with a single term in the specification.

* A trace in which a series of observations correspond to a single term in the specification as a result of iteration or recursive process definition within the specification.

* A trace in which a number of observations results from a number of concurrent processes, created by behaviour within the specification, each of which has its own behaviour leading to the action of interest.

* A trace in which a number of observations result from the activity of a number of different systems, each of which individually corresponds to the behaviour specified.

* An assertion that a particular system will be observed to behave in the specified way once.

* An assertion that, whenever certain preconditions are set, the system will be observed to behave in a certain way, either in sequence or concurrently.

* An assertion that any one of arbitrarily many systems to which the specification applies will be observed to behave in a certain way whenever the preconditions apply.

Thus we have specifications that apply to large groups of identical systems (such as mobile-phones), to many uses of unique systems (such as a bespoke telephone enquiry system), to one specific use of one specific system (such as the system that will control the sequence of events during the opening of the Olympic Games). We have historical observation and claims of conformance for future behaviour. Orthogonal to all these, we have instances of the same action type within the specification.

One might expect a general-purpose specification technique to be able to cope with all of these different cases. However, great care must be taken when expressing constraints on the structure of the internal structure of specifications not to imply unintended limitations scope. As a case in point, a clarifying note about the interpretation of the term "action" within an ODP specification, intended to distinguish between differences between the terminology used in previously standardized specification languages, has been read subsequently as implying a restriction of scope, causing much confusion.

It is important to be clear exactly what interpretation of the correspondence between specification and real world is intended, because in practice specifications are neither static nor monolithic. This adds a further potential complication when specification fragments are reused or changed during a system's lifetime, and it is here that policy becomes involved. One of the reasons for identifying specific policies is to tailor generic specifications to the specific, dynamically changing, situations in which they are to be used.