Finland votes "YES" for the progression of Part 1 to Committee Draft, with the following comments:

General

Even though RM–ODP Part 1 still needs quite a lot rewriting and even some restructuring, Finland votes YES for its progression to Committee Draft. The first clauses up to Clause 10 are already pretty readable and Finland has only few comments concerning them. The rest of Part 1 is more immature and fragmented and more changes are proposed to make it more comprehensive and more useful for all who want to understand and apply RM–ODP.

Major areas of concern are:

– A more useful introduction of ODP object model explaining the main concepts of ODP modeling and stressing the difference from the traditional OO modeling is needed.

– The enterprise language chapter needs rewriting.

– In the chapter about computational language the binding model description is not appropriate neither the clause about subtyping.

– The whole chapter about engineering language is very fragmented and seems to discuss things in pretty random order. It should be totally rewritten.

– The chapter about ODP functions needs some rewriting.

– Transparencies should be described before functions.

Detailed comments

Item: 1
Front page
Comment:

The name has changed to Open Distributed Processing – Reference Model – Part 1: Overview

Item: 2
Introduction, Page vi
Rationale:

The names of part 2 and part 3 have changed. The first six paragraphs of the introduction should be same in all parts of the ODP Reference Model Standard.

Proposed action:

The text should be copied from Part 3 Introduction.

Item: 3

Clause 2.1, page 1

Rationale:

The names of part 2 and part 3 have changed.

Proposed action:

Names of the Part 2 and Part 3 should be updated.

Item: 4

Clause 3, page 2

Proposed action:

List of bibliography can be deleted.

Item: 5

Clause 4.2, page 3–4

Rationale:

The list includes terms context, framework and observable action that are not defined in Part 2. It lacks some Part 2 terms (liaison, environment contract) that are used in Part1. Relocation transparency is missing from the list of Part 3 terms. Also many other terms from Part 3 are used in Part 1.

Proposed action:

Update the list.

Item: 6

Clause: 8., 10., etc

Rationale:

It is rather misleading to introduce ODP modeling technique as an object oriented method. There is some fundamental differences between traditional object modeling techniques or languages and ODP–object models. It should be clarifying to describe what these differences actually are.

Proposed action:

Stress the importance of roles, not the importance of objects.

List the differences to traditional object models:

- objects change from viewpoint to viewpoint, it is not only a decomposition process when the viewpoint is changed,
- each viewpoint looks at the whole system, but on different abstraction viewpoint,

- the function under specification does not describe the whole software, instead a group of functions (specifications) have to be assigned to the software system,
- type in ODP model is only looking at one specific property at a time, it does not define anything about the existence or lack of other properties,
- information viewpoint schemata map to the traditional object model vocabulary in the following way:

ODP invariant schema	OO information consistency constraint,
	OO static schema
ODP static schema	states of the information storage,
	OO dynamic schema
ODP dynamic schema	OO operations

Item: 7
Clause: 8.2 Viewpoints
Rationale:

The current text does not give clear enough picture about the purposes of the viewpoints, especially "roles" of computational and engineering viewpoints remain unclear.

Proposed action:

Change the text to following:

8.2 Viewpoints

The five viewpoints defined in the ODP Reference Model can be divided into three classes:

a) Viewpoints giving an external description of the target system (the enterprise viewpoint)
b) Viewpoints giving an internal specification of the target system (the information viewpoint and the computational viewpoint)
c) Viewpoints giving a description of the use of the underlying infrastructure on which the target system can rely (the engineering viewpoint and the technology viewpoint)

The environment of a system has always requirements for the system. The system is there to fulfill these requirements: there is a mission for the system. The enterprise viewpoint focuses on the purpose, scope and policies for the the system. It describes the position of the system in its environment and describes its mission. It points the roles played by the system, the activities undertaken by the system and the policy statements about the system.

The information viewpoint and the computational viewpoint specify the system. They show the methods used in the system to the mission described in the enterprise viewpoint.
The information viewpoint is concerned with the semantics of information and information processing in the system. It specifies the configuration of information objects and the behavior of those objects.

The computational viewpoint is concerned with actions which change the information in the system. These actions are performed by computational objects: the system is specified as a configuration of interacting objects, without concerning how the interactions between objects are facilitated. Computational objects rely on the infrastructure to provide suitable means for their interactions.

The engineering viewpoint describes how the system uses the infrastructure that supports transparencies for distributed computing. The engineering description specifies how to provide the transparencies needed by the computational system – it points out the requirements for the infrastructure.

The technology viewpoint is concerned with the details of the components from which the distributed system is constructed.

Item: 8
Clause: 8.3 Distribution transparency
Rationale:

The first paragraph seems to be here only to explain again why distribution transparency is needed.

The second paragraph would be more logical if the order of the first two sentences is reversed.

The last paragraph (" In a centralized ..) of this chapter seems to be a bit out of place here.

Proposed action:

Remove the first paragraph (starting with " ODP standards aim to enable ...").

Change the order of the first two sentences in the second paragraph so that it starts with the second sentence(starting with "Distribution transparencies ...." followed by the first sentence (starting with " The ODP infrastructure supports a set of ...").

Move the last paragraph to Clause 10.2 Contract.

Item: 9
Clause 8 Summary of the RM–ODP
Rationale:

For clarification. There has been some difficulties to understand the relationship of the infrastructure and its components, especially when specifying new ODP functions.

Proposed action:

After the clause 8.3 add a new clause 8.4 (and chance the existing clause to 8.4 Conformance ...) describing the ODP systems and the infrastructure with following text:

8.4 ODP systems and the infrastructure

An ODP system consists of a number of applications, which all can be specified as separate (sub)systems. These (sub)systems (the target systems for the specification) use support services provided by the infrastructure selectively, depending on the needs of the (sub)system. There might be several target systems running on the same infrastructure.

All the target systems relying on the infrastructure can have their own requirements for the infrastructure. The ODP infrastructure can be seen as a separate target system fulfilling this set of requirements, which is the union of the requirements by the individual target systems.

The functional decomposition of a target system is specified in distribution transparent terms. The target system specification relies on the infrastructure to hide the distribution to the degree wanted.

The distribution transparent services provided by the infrastructure can be achieved by ODP functions. The infrastructure consists of ODP functions, which all are separately specified as normal (sub)systems.

Item: 10
Clause: 10.1.1. Objects
Rationale:

Corrupted definition: " ... the ODP system consists of itself and its environment ..."

Proposed improvement:

" ... the ODP system consists of *the object* itself and its environment ..."

Item: 11
Clause: 10.1.1.1
Rationale:

This does not say anything about abstraction. It should discuss roles as a method of abstraction.

Roles do not appear early enough.

All descriptions of systems talk only about roles, i.e., mission related abstractions of the object, omitting all management, security, etc., relate properties of the object.

Proposed action:

Abstraction and roles should be explained.

Item: 12
Clause: 10.1.3 Object modeling in ODP
Rationale:

The purpose of the last two dots in the list is not quite clear.

Question:

Do other object models constrain the capsulated behavior of objects or the type of their interactions?

Item: 13
Clause: 10.2
Rationale:

It is unclear whether the words "specification language" refers to languages like Lotos and SDL, or to viewpoint languages.

Proposed action:

Clarify what is meant by specification languages here.

Item: 14
Clause: 10.2.2 Templates, page 13
Comment:

The text about behavior compatibility is somewhat unclear.

Item: 15
Clause: 10.2.3. Role, page 13
Rationale:

This definition of role refers to a very technical detail. In interface signatures, a role identifies which action the object on that operation is willing to take – an initiator role or a role of responding object. This view to role is not interesting at the Part 1 description. Instead, the concept of role appearing at the enterprise viewpoint language is more important.

Proposed action:

Replace the definition of role following text: The behavior of an object can be partitioned into subsets which perform some specific functions. Such subsets can be named as roles in the RM–ODP. When an object is viewed in a role, only a named subset of its actions is of interest, and other actions are abstracted away – possibly to other roles. Interactions (i.e. actions that communicate with the environment of the object) of a role can be grouped together to an interface. ODP objects usually have many roles. Every object has its normal functional or mission role i.e. the purpose it was designed for and for management purposes it has a management role, the behavior needed to monitor and control the mission role behavior. Both roles have their own interfaces, the mission role offers a mission interface and the management role offers a management interface.

Item: 16
Clause: 10.3.
Rationale:

Other important examples of the group concept are community and federation. They should also be explained here, even when they are Part 3 concepts.

Proposed action:

Explain domain, community and federation as subclauses of general group concept: 10.3.2.1 Domain, 10.3.2.2 Community, 10.3.2.3 Federation.

Item: 17
Clause: 10.3.4. Contract
Rationale:

The second sentence in the first paragraph is a bit misleading in stating that " a contract constrains the cooperation of objects".

Proposed action:

Replace the second sentence in the first paragraph by "A contract is an agreement that enables the cooperation between a set autonomous objects by regulating the behavior of the participating objects."

Item: 18
Clause: 10.3.4. Contract
Rationale:

The concept of obligation has caused confusion several times. It should be described more clearly.

Proposed improvement:

Replace the third sentence in the first paragraph (" A contract can ...") by "A contract can specify the roles of objects, the expected behavior of the objects (by prescribing the required behavior i.e. obligations of objects, the allowed behavior i.e. permissions of objects and/or the forbidden behavior i.e. prohibitions of objects), and the quality of service aspects of the cooperation (e.g. deadlines, throughput, reliability, security). A contract can also state the time it is valid and indicate what kind of behavior invalidates the contract.

Item: 19
Clause: 10.3.4. Contract
Rationale:

This clause should contain also something about an environment contract.

Proposed action:

Add some text explaining environment contract. The last paragraph (" In a centralized ..) of Clause 8.3 (Distribution transparency, page 9) could be moved at the end of Clause 10.3.4.

Item: 20
Clause: 10.3.5
Rationale:

In this clause there are two incorrect statements: the second sentence of the clause "Binding is the preliminary phase of contract establishment." and the fifth sentence "Binding yields a liaison."

Proposed improvement:

Remove the incorrect sentences. Add after the fourth sentence (ending " ... obey the contract") "Some kind of specific establishing behavior is necessary to put a contract in place between objects, likewise a specific behavior for terminating the contract. Binding behavior is the establishing behavior that makes it possible for objects to interact. Binding, the result from binding behavior, is an example of liaison.

Item: 21
Clause: 10.3.5.
Rationale:

Explicit and implicit binding are not described.

Proposed action:

Add a short description about explicit and implicit binding.

Item: 22
Clause: 11.1.
Rationale:

The second paragraph uses terms interacting agents, but agents are no more enterprise language concepts.

Proposed action:

Replace agents by objects.

Item: 23
Clause: 11.1.
Rationale:

This text does not give any guideline for writing a specification. Although, it should give an easily read instruction.

a) things in dot−lists are too detailed
d) organizational rules:
   objects should not appear in this way in the enterprise viewpoint specification

rest of the d–point is not understandable

e) wrong level of abstraction

f) not understandable

Definition (end of page 17) "performative action is an action that changes policy" is incorrect.

Proposed action:

Rewrite the Clause 11.1 to be more readable, to reflect the new Part 3 and to give better guideline for writing a specification.

The structure of the chapter could be.

* Purpose : function objectives
* Roles
* Community
* Activities
* Policy statements

Item: 24

Clause: 11.2

Rationale:

The first bullet list requires more from an information specification than does Part 3.

Implementation conformance is described only in information viewpoint. This can lead to think that

a) conformance is only meaningful in information viewpoint

b) only implementation conformance is of importance

Proposed action:

Delete two bullets from the list: the third (demanding quality attributes for information) and the last (about objects created for the convenience of implementors).

Delete also the sentence after the first bullet list (starting " An information specification defines the classes of concepts ....").

Either delete the text about conformance or extend it to include other types of conformance and include similar paragraph about conformance to other viewpoints.

Item: 25

Clause: 11.3

Rationale:

Incorrect statement: " ... details of the infrastructure supporting these interactions are described in the engineering viewpoint..." It is true only if the system described happens to be a communication system. The description should contain application specific function details.

Proposed action:

Delete "details" from the sentence: "The infrastructure supporting these interactions is not visible in the computational viewpoint. It is described in the engineering viewpoint. "

Item: 26
Clause: 11.3.1. Computtional interfaces, page 19
Proposed action:

Correct the typo in the heading

Item: 27
Clause: 11.3.1.
Rationale:

Operational interface does not describe sequences of operations, but only the possible components of those sequences (the first dot). The distinction of templates and instances was not clear in Part 3, so at least this text should make it clear, instead of following the wrong interpretation.

The last sentence in the second dot about stream interfaces " Its signature contains the type of the flow and an indication of causality (e.g. direction of flow)." is a bit confusing. A stream usually has many flows and the sentence hints that causality could be something else than just the direction of bit stream.

Also the real usage of signatures remains a bit unclear.

Proposed action:

Replace the text for the first dot by following text: The operational interface is an interface whose signature describes the operations in the interface and the causality of the interface as a whole. An operation is either an announcements or an interrogation. Each operation type in the interface is described by an action template that specifies the name of the invocation and the parameters of the operation. For interrogations also the possible terminations and their parameters are specified in the action template.

Replace the second dot about stream interfaces by following: A stream interface describes an interface where all interactions are flows, i.e. sequences of interactions resulting from moving information, e.g. audio or video, from a producer object to a consumer object. Its signature contains an action template for each flow type in the interface, indicating the causality of the flow type (i.e. producer or consumer).

Add text explaining the usage of signatures.

Item: 28
Clause: 11.3.2

Rationale:

The clause is very detailed and totally out of place.

Proposed action:

Remove the existing clause 11.3.2. If regarded necessary write some new text about subtyping.

Item: 29
Clause: 11.3.3 – 11.3.5. Binding
Rationale:

Binding concept has been changed in Part 3.

Proposed action:

These clauses need rewriting to reflect the changes and also to be more readable.

Item: 30
Clause: 11.4 Engineering language
Rationale:

The whole chapter about engineering is very fragmented and seems to discuss things in pretty random order.

Proposed action:

Rethink and restructure the chapter about engineering language; what are the subjects that should be discussed here and in what order.

Item: 31
Clause: 11.4.1 Editor's note
Rationale:

The current Part 3 gives answers to these questions.

Proposed action:

Delete the notes.

Item: 32
Clause: 11.4.1 Correspondences between...... , page 25, 26
Rationale:

The text is no more quite valid, it has many mistakes and it is not very clear. And it is a bit odd to have correspondences between computational and engineering specifications as the first thing about engineering language. Part of it (starting from the second last paragraph on page 25) deals more with the structure of channel than with correspondences between specification. And for some reasons the interceptor is left out of the channel.

Proposed action:

The text should be rewritten totally. Add following text explaining the correspondence between computational and engineering specifications in a suitable place after the restructuring: "The computational specification describes, in certain detail, the functionality of the system as interacting computational objects. The engineering specification is constrained by the computational specification so that it has to respect the computational objects and their interfaces. Each computational object must be represented as one basic engineering object or as a group of basic engineering objects and for each computational interface there must be one corresponding engineering interface. In an engineering specification it is not allowed to combine two COs into one BEO or to represent a group of COs by a group of BEOs that violate the boundaries between COs.

An engineering specification is not at all interested in the mission functionality of the ODP system (that is totally given in the computational specification). It focuses on distribution of BEOs and on selective hiding of the effects of distribution from the BEOs (and from the users of the ODP systems). In the computational specification it is not necessary to consider the problems caused by distribution, in the engineering specification it is not necessary to consider the mission functionality. The engineering specification concentrates on describing the necessary infrastructure for providing of processing services to BEOs in spite of the distribution. The infrastructure includes objects (= objects in the channel) that facilitate interaction between distributed BEOs, objects that provide different kinds of distribution transparencies, objects that control and manage other objects and their interactions etc.

The basic assumption is one to one correspondence between COs and BEOs. For reasons of performance, security, reliability, availability etc. a group of BEOs can be needed for one CO.

Item: 33
Clause: 11.4.2 Channel establishment
Comment:

The third paragraph on page 27 ("The type of an engineering ...) talks about interceptors, but interceptors are introduced first on page 29.

Item: 34
Clause: 11.4.3 Management interfaces
Rationale:

The text of this clause is very general and does not explain management interfaces at all.

Proposed action:

The text needs rewriting

Item: 35
Clause: 11.4.4 Federation
Rationale:
Very general text about federation. Part of it (on page 30, three paragraphs starting from with "Multi−hop addressing ...) is very unclear.
Proposed action:
The whole clause needs rewriting.

Item: 36
Clause: 11.6 Consistency between viewpoints
Rationale:
Correspondences between computational and engineering specifications are explained already in the clause 11.4.1.
Question:
Would it be better to move here those parts of the rewritten clause 11.4.1 that are explaining correspondences?

Item: 37
Clause: 11.6 Consistency between viewpoints
Comment:
The third paragraph promises that detailed consistency rules are found in Part 3. Actually the rules are not very detailed, but rather general.

Item: 38
Clause: 11.7, 11.8 functions, transparencies
Rationale:
We feel that transparencies should be described before functions as transparencies are the main reasons functions are needed for. Describing first functions and after them transparencies feels like reasoning backwards: first you tell what they are, and only later on why.
Proposed action:
Clause 11.7 should be about transparencies and clause 11.8 about functions.

Item: 39
Clause: 11.7.1 Management functions
Rationale:
The text about management functions is not easy to read. It is pretty condensed and does not give much explanation (e.g. pictures, examples). Also application management is not included at all. Besides there are also some sentences we do not understand. The last sentence of 11.7.1.1 " Node−level scheduling is optimized if message arrival pre−emts computation." seems to be out of place here. It is not quite clear what is meant by the two last sentences on the page 34 claiming that "checkpointing policy and storage service for a cluster are embodied in the cluster itself; it may decline requests from unauthorised clients" It is the cluster manager

that embodies the management policy and here it is more a question about security policy than checkpointing or storage.

Proposed action:

Rewrite clause 11.7.1 Management functions to be more readable and to include application management.

Item: 40

Clause: 11.7.2 Coordination functions

Rationale:

Under coordination functions there are only two functions explained: the replication function and the transaction function.

Question:

Are these the most important /difficult coordination functions that need to be explained in detail and other functions so simple/unimportant that there is no need to mention them?

Item: 41

Clause: 11.7.3 Repository functions

Rationale:

Only one repository function (the relocation function) is described without any introducing text.

Proposed action:

Write a short introducing text describing why repository functions are needed and what kind of repository functions are offered by the RM–ODP. Then explain in more detail those repository function that need to be described because, e.g. the Part 3 description is not clear enough or the usage of the function needs more explanation.

Item: 42

Clause: 11.7.3.1 Relocation function

Rationale:

The text in this clause is misleading. It gives the impression that all reactivation is controlled by relocators and that all interfaces must have relocators.

Proposed action:

Rewrite the text (or delete it if regarded that the relocation function is described well enough in Part 3).

Item: 43

Clause: 11.7.4 Security functions

Comment:

Includes all other security functions listed in Part 3 except key management function.

Item: 44
Clause: 11.8 ODP distribution transparency
Rationale:

The first sentence in the last paragraph on page 37 starting with "Distribution transparency is ....." is not quite correct in stating that distribution transparency hides properties of distribution from specifiers only. Part 2 states that distribution transparency hides properties of distribution from users that include end–users, application developers and function implementors.

Proposed action: Correct the sentence.

Item: 45
Clause: 11.8 ODP distribution transparency
Rationale:

All over in this clause is stated that " ... transparency hides from basic computational object .... ". Basic computational object is no more used in Part 3 and hiding is not constrained to computational objects.

Proposed action:

Replace a basic computational object by an object (when it fits with the Part 3 definition of the transparency).

Item: 46
Clause: 11.8.1 Access transparency
Rationale:

The last three paragraphs in this clause are not very comprehensive. They seem to deal with some details of access transparency.

Proposed action:

Either add some sentences to tie the last three sentences to the the beginning of the clause or delete these three sentences.

Item: 47
Clause: 11.8.6 Failure transparency

Rationale: The explanation of the failure transparency in the first sentence of the clause (" ... either prevents a basic computational object from failure or hides that failure from itself") is not quite according to the definition of failure transparency in Part 3.

Proposed action:

Replace the explanation by the following text: "Transparency hides, from an object, the failure and possible recovery of other objects or the object itself, to enable fault tolerance.

Item: 48
Clause: 13.1., 13.2, 13.3.
Rationale:

Part 2 gives the management domain as an example of specialized domains.

Part 2 also uses the term controlling object instead of coordinator object.

Proposed action:

Domains should be replaced by management domains and coordinator objects by controlling objects.

Item: 49

Clause: 16.6

Comment:

This example system is now introduced both in the Trading function draft and here. One place would be enough. Anyway this functionality is going to be part of the NWI.

Item: 50

Clause: Whole Part 1

Comment: Schemata should be used as the plural for schema as is done in Part 3.