Finland votes "yes" for the progression of Part 2 to Draft International Standard, with the following comments:

A. Some minor editorial comments

A.1 Page ii, Introduction, 2. par. , 1. line

Replace X.901 to X.905 by X.901 to X.904.

A. 2 Page ii, Introduction, 5. par. , 4. line

Replace the sentence starting "It also contain a list of required areas .... " by the sentence "It also contains a categorization of required areas of standardization."

A.3 Page ii, Introduction, 8. par., 1. line

Remove ", specification techniques and formalisms".

A.4 Page 6, Clause 9.9 Subtype/supertype, Note

The note states that the object template subtype/supertype relation does not necessarily coincide with behavioural compatibility. The same thing is repeated on page 8 in clause 9.19 Template type, in 2. paragraph. Is it really necessary?

We think that the note from clause 9.9 could be removed.

A.5 Page 10, temporary note below the picture

Remove it (and all other temporary notes).

B. Minor technical comments

B.1 Page 5, Clause 8.3 Action and Clause 8.4 Interface

It is not always made clear in the definitions when interactions actually mean action templates and when instances of action templates. In many cases it does not matter, but sometimes it causes some difficulties in understanding. E.g. the interactions in clause 8.3 seem to be templates and in clause 8.4 the interactions of an interface seem to be instantiations of action templates.

B.2 Page 10, Clause 10.1 <X>–Group

The examples given in the Note are too specialized and therefore a bit misleading. We think that the note should stress that there are at least two different kinds of groups

* groups that are in fact domains
  – communication group / protocol group

– access group
– technology group
– failure/fault group
– node, capsule, cluster (object groups) / location

* groups that are in fact communities
– server group / replica group
– trading federation group (shared objective: trading)
– transaction group
– interaction group (binding)

B.3 Page 10, Clause 10.2 Configuration

The configuration concept is not very general concept. We feel that it is more a concept belonging to computational or engineering viewpoint.
Remove configuration from Part 2 .

B.4. Page 10, Clause 10.x

The community and federation concepts that are defined in the enterprise viewpoint of Part 3 should be moved to Part 2 .

Rationale:

Part 2 and Part 3 define terms "domain", "community", and "federation". However, there seems to be different views in Part 2 and Part 3 about the usage of these terms. This has caused problems for example in Trading function group.

The following interpretation has been used:

DOMAIN
Domain is an area in a technical sense. This area is restricted by a shared rule. In ODP−language it could be said that everything in a domain is of same type. Domain could be for example a security domain, where everyone is allowed or denied same service. There could also exist a communication domain, where everybody uses the same protocol (== "protocol domain").
Domain does not restrict any community issues.

COMMUNITY
Community is a group of members with a common activity. Everybody in a community work for a shared goal. Community members may be situated at different domains.

FEDERATION
When objects at different domains communicate, it is called federation. For example, two objects in a security domain may need an interceptor (protocol translator) because they are situated on different communication domains (they

use different protocol). Federation means crossing domain boundaries, one or many. Each domain boundary requires an interceptor (pair).

INTERWORKING
When objects at different communities communicate, it is called interworking. Interworking means, that objects already playing some roles within their communities assume additional roles that bring them into an additional community.

Usually interworking or cooperation is referred to when an application is distributed. The objects in the distributed system play the same role towards the local clients, but when they have to synchronize or use other distribution dependent features, they assume additional roles.

B.5 Page 11, Clause 11.3.2 Isochronity

The definition of isochronity seems to be too technical and too restricting when demanding that " every adjacent pair of actions in the sequence occupy equally−sized intervals of time".

B.6 Time concept

Time concept is not defined in RM−ODP. However the concept is used in several situations (e.g. in 8.7, 8.10, 8.11, 11.3.2).
Either time should be defined based on causality concept or the references to time should be replaced by references to causality concepts.