FINNISH DATA COMMUNICATION ASSOCIATION STY

UNIVERSITY OF HELSINKI

TITLE: Comments on SC21/WG7 N897

SOURCE: Finland

PROJECT:

STATUS: National body comments

DATE: June 1994

NOTES: Finnish National Body contribution, to be discussed at the joint SC21/WG7 meeting in Southampton

Introduction

Trading function document should reach CD level at the Southampton meeting. To achieve this goal, the most important thing to be resolved, is the issue on federation. Leaving federation as a separate standard is not an acceptable solution. In other areas, basic structure of trading function has almost reached maturity for CD. However, several statements about scope of trading function utilizability and policies within trader community need consideration.

In regard to federation, syndicate is a unnecessary concept. It was a useful term to use in the Turin meeting –– progress was reached on areas not involving federation. Federation aspects should be considered as an integral part of the trader object itself, not seen as an extension requiring separate functionality. In our detailed comments, we try to broaden the scope of already accepted trader operations in such a way that they also apply in case of federation. Our suggestions are close to those made by UK in the Turin meeting.

The tutorial and introductional parts of the drafted standard give an impression of aimed usage In our opinion, this implied usage is partially too narrow and partially contradictory with other parts of the draft. Reading the draft gives the impression that trading can be used only to support a middleware layer structure within an ODP engineering and computational viewpoints. If this is the aimed usage, the policies, QoS etc, should be prescribed in this standard. However, trading does have more application areas. Systems that have almost identical structure as the ODP trader are already used, for example, for information management purposes. Such systems need a different set of allowed services, allowed policies, allowed QoS, etc, as middleware systems. Our comments on trader usage areas aim to broaden the discussion in tutorial and introductory parts of the draft.

Policies within the trader community do already in the current draft imply several cases where trading community or trading syndicate borders are crossed. The policy infrastructure is unclean and does not follow principles of orthogonality. Our comments try to ortogonalize the policy area: we try to separate federation oriented policies into a few rules, and try to keep other rules strictly within one trading domain only. It may be helpful to notice that several traders may have same owner, which may cause the traders to follow the same policy rules. Although traders are autonomous in each domain, they may be forced (by the owner) to follow *shared* policy.

Our proposals are imbued with three ideas:

- Objects can change roles, and they can participate in several communities, simultaneously. Also, forming a community is only a question of playing roles.
- Trading is an ordinary service, just like printing.
- Community is not a concrete, stable thing –– it's only a view to objects participating in a common activity. A trader can simultaneously participate in two communities: In one community a trader is serving clients and servers. An other community consists of trader objects in client, server and trader roles. In this latter community the objects act in behalf of the communities of the first type.

Overall motivation for these comments has been introduced in the Finnish national body contribution "Federation transparency in the ODP Trading function", June 1994.

Detailed comments

Item: 1
Clause: 4.1, 4.2.
Importance: Editorial
Rationale:

Part 2 and Part 3 have been modified by Geneva meeting.

Proposed Improvement:

Remove references to cardinality constraint, integrity rules, resource manager, partition, policy maker, and relation. These concepts have been removed from Part 3. Clause numbers in references should be changed according to Part 2 and Part 3 Southampton output documents.

Item: 2
Clause: 4.1., 4.2.
Importance: Editorial
Rationale:

Concepts of trading, exporter, importer, import and export policy, service import, service export, federated export, trading community, trading federation and trading policy are defined by Trading function standard and from it adopted by standards X.902 and X.903. It is misleading to reference to the definitions of these concepts in X.902 and X.903, because those always are "a round behind".

Proposed Improvement:

Remove above mentioned references from trading function standard draft, and replace them with full definitions. If this is done, the trading function working group should then proceed by suggesting that Part 2 and Part 3 removes definitions of these concepts and replaces them by references to Trading function standard.

Item: 3
Clause: 5
Importance: Conceptual
Rationale:

Part 2 and Part 3 describe the concepts of object, contract and interaction. Contract and a contractual context of objects (liaison) seem to be the fundamentals. Furthermore, Part 2 says in 13.4.5:

"Trading is the interaction between objects in which information about new or potential contracts is exchanged via a third party objects. It involves:

a) exporting: the provision of an identifier to an interface which is claimed to meet some statement of requirements (i.e.*offer a potential contract*);

b) importing: the provision of an identifier to an interface which matches a given statement of requirements, allowing a future binding behavior to take place (i.e. the establishment of a contract)."

From these ideas, a new description for trading function can be drawn. Trading is exchanging information about contracts.

When exporting, a server object gives out information about itself. It declares that it is in such a context that it is willing to establish a contract with a client fulfilling some predicates. Thus the exporter establishes a contract with the trader: both the trader and the exporter knows that a potential contract is stored into the trader. Trader and exporter are in liaison with each other.

For importing, client and trader have to be in liaison with each other. Trader must allow importing from it and the client has to know that the trader exists and is able to help in service offer matching.

To the client, importing a service offer is part of contract establishing behavior with a server object. First, the client does not know with whom to start the behavior. It requests the trader for suggestions. In the import operation the client receives a (set of) potential contracts. It becomes aware that there is an interesting server object that is in context where establishing behavior stated by the traded contract is possible. After importing is completed client and server are in liaison with each others. Although the server does not know the identity of client (yet), it knows that the client fulfills the required predicates, so that liaison can take place.

Liaison may lead to establishing a binding between client and server objects.

Proposed Improvement:

Change text in chapter 5 so that it discusses potential contracts by defining service offers.

Service offer:

A contract template that exporting object publishes. It contains predicates restricting the domain of importers with which this contract can possibly be established and description of the properties and behavior of the server object itself.

To accompanion the service offer, service request (importing) is needed.

Service request:

A contract template that importing object publishes. It contains predicates restricting the domain of exported contract templates considered. Specially it may contain demands about the suitable behavior offered.

Item: 4
Clause: 5
Importance: Editorial
Rationale:

Trading function is by nature such that it brings together several autonomous areas or domains. Thus, trading function should be viewed as a "global" activity, not a domain based activity. Therefore, in the introduction it seems unnecessary strict to use a separate chapter for federation issues.

Proposed Improvement:

Remove the title 5.1.
Move the text of 5.1. before the three−bullet−list on page 5.

Item: 5
Clause: 5
Importance: Technical
Rationale:

Syndicate was a intermediate working concept for Turin meeting. Text about interworking among traders should be modified so, that syndicate is not needed.

Proposed Improvement:

Replace chapter 5.1. with the following:

Due to the sheer number of service offers that there will be in the world and the differing requirements users will have of a trading service, it is inevitable that the service will be split up and the service offers partitioned. Each trader has a set of importers and exporters that it provides services to. This set, including trader itself, is called trading community. Trading community may be formed because of common geographical location, common technology, common policy, common administration or common set of provided or needed services. None of these reasons forces to a community boundary.

Even if service offers are partitioned, users will still require access to services advertised in different partitions. Traders may with to interact with each other to expand the set of service offers they provide to their importers. Consequently, there is a need to discuss federating traders.

Item: 6
Clause: 6
Importance: Editorial
Rationale: See item 3.

Contract (as defined in Part 2, 11.2.1): An agreement governing part of the collective behavior of a set of objects.

Proposed Improvement:

Replace line 1: A trader provides the means to advertise and match service offers. A service offer is a contract template published by an object. A contract template specifies, what is the behavior of this publishing object itself.

Item: 7
Clause: 6.1.1
Importance: Editorial
Rationale:

Inconsistency with text in 6.1. and 6.2, or Part 3.

Proposed Improvement:

Trading community: Remove trading policy maker, internal arbitrator.

Change export policy to exporter policy.

Add importer policy controller.

Item: 8
Clause: 6.1.1
Importance: Technical
Rationale:

Resolution of federation issues.

Proposed Improvement:

Remove text about trading syndicate.

Change text about trading federation to following:

Trading federation is an activity between traders. Traders may form a trading community, where trader objects take the roles of importer, exporter and trader.

Item: 9
Clause: 6.1.2
Importance: Editorial
Rationale:

Inconsistency with text in 6.1. and 6.2, or Part 3.

Proposed Improvement:

Replace 6.1.2.3, 6.1.2.4 and 6.1.2.5 by:

Trading administrator – an agent enforcing set of services traded, restricts the members of trading community, selects the policy governing the participation in federation with other traders and governing the policy used in the process of matching service requests and service offers.

Note: Trading administrator may be a community of administrators, each enforcing one or more policies. For example, type management administrator and global search policy administrator may be separate.

Trader owner – an agent or artefact defining the policies set in effect by trading administrator.

Note: Trader owner may be a community collectively making the decision about different aspects.

Item: 10
Clause: 6.1.2.8
Importance: Conceptual
Rationale:

An agent may initiate actions, an artefact may not. Trader never initiates an action, unless in act in the role of exporter or importer in a federation community of traders. Trader in trader role within the federation community does not initiate actions.

Proposed Improvement:

No actions taken within the trading function standard yet. However, the trader group could request the Part 3 group for clarification on the intended differences between the artefacts and the agents. If objects only take roles, their may be passive in some interactions and actively initiate some others. Why the separation is then needed? It the separation has to be done, how does the Part 3 group classify objects in trader role?

Item: 11
Clause: 6.1.2.8
Importance: Conceptual
Rationale:

Modeling of trader object as a set of trader properties, interfaces composed of operations and a storage.

Proposed improvement:

Replace 6.1.2.8 by:

6.1.2.8 Trader

Trader is an agent within a trading community which

– participates into the activities of service exporting and service importing
– has the responsibility of storaging the service offers
– is governed by trader properties.

Trader properties include trader policies, locations of trading interfaces, reference to the service type management function used, etc.

Item: 12
Clause: 6.1.4.1
Importance: Conceptual
Rationale:

Activity is a graph of actions. Actions should here be something of interest for the enterprise viewpoint to trading community or of trading function. Performative action change obligations, prohibitions and permissions of objects.

An server object and a trader object can establish a liaison with each other. In this liaison, the trader object has knowledge about the server object (service offer) and is permitted to expose this knowledge to other object (limited by the policy given by the server object). Also, this liaison can be terminated by taking some action.

The establishment of the liaison must clearly involve both objects. However, the termination of the liaison can happen in different ways. For example,

– by an terminating, observable interaction (withdraw)
– by internal actions within the objects in liaison, based on an earlier agreement (expire date).

Service exporting is a chain of actions taken by the objects in the roles of trader and server. The chain of actions establish and terminate a liaison in which the trader object is allowed to expose knowledge about server to a limited group of clients.

Service importing is a chain of actions in which the objects in the roles of client and trader communicate. The client receives information about server objects in the trading community.

Proposed improvement:

Included into the proposal in item 13.

Item: 13
Clause: 6.1.4.1 and 2
Importance: Editorial
Rationale:

See proposal on item 12.

Liaison is a state of having a contractual context in common. Thus the liaison must be specified by telling on what things the objects in liaison have actually agreed.

Look item 3 for definitions of service offer and service request.

Proposed Improvement:

Rephrase 6.1.4 as:

6.1.4. Activities

Trading community has two activities, service exporting and service importing.

6.1.4.1. Service exporting

Service exporting is a chain of actions taken by the objects in the roles of trader and server. The chain of actions establish and terminate a liaison in which the trader object is allowed to expose knowledge about server to a limited group of clients. The liaison is characterized by a service offer.

6.1.4.2. Service importing

Service importing is a chain of actions in which the objects in the roles of client and trader communicate. The client receives information about server objects in the trading community. The communication is characterized by a service request from the client to the trader and a set of corresponding service offers from the trader to the client.

Item: 14
Clause: Editor's question on page 7
Proposal:

Trader creation is taken as a general object instantiation process. In information viewpoint description the trader template and its initial state must be described.

Item: 15
Clause: 6.1.5.1, 6.1.5
Importance: Editorial
Rationale:

Trading policy is only used at this clause. It gathers together all policies within the trading community. Trading policy and trader policy is conceptually very different, but textually so close to each other, that the difference is very easy to miss.

Proposed Improvement:

Remove 6.1.5.1 and the usage of "trading policy" term. In case, all policies of the trading community must be referred, it is much more safe and readable to use longer phrasing.

Retitle 6.1.5 as "Trading community policies".

Item: 16
Clause: 6.1.5.4
Importance: Technical
Rationale:

Need of orthogonality of policies and need of federation oriented policies. Read the accompanying contribution "Federation transparency in ODP Trading function" for further reasoning.

Proposed Improvement:

Replace 6.1.5.4.:

Trader policy – a set of policies in trader that govern the traders activities. This set includes

- service offer acceptance policy
- storage policy
- import request acceptance policy
- import action policy, consisting of
  - trader assignment policy
  - local search policy
  - resource consumption policy
  - service type federation policy
  - global search policy
- trader management security policy.

Item: 17
Clause: 6.1.5.5 Trader security policy
Importance: Editorial
Rationale:

Current text implies that management interface are not secured at all.

Proposed Improvement:

Replace by:

Trader security policy – A security policy of an object restricts the set of objects that can interact with it. Trader security policy is a combination of service offer acceptance policy, import request acceptance policy and trader management policy.

Item: 18
Clause: 6.1.5.8
Importance: Technical
Rationale:

Traders are autonomous. A trader may not force another trader to accept an exported service offer from different domain. However, a trader may advertise a service for exporting to other traders. In the trading community such a service can be available.

Proposed Improvement:

Remove 6.1.5.8.

Item: 19
Clause: New
Importance: Technical
Rationale:

Trader needs an internal policy for storaging the offers in a optimal way. The policy states in which way the offer space within one, autonomous trader is partitioned. There is no obligation to partition the space, or prescriptions how the partitioning should be done. As a tutorial material outside the standard different optimization methods for storaging structure are interesting to study, but they do not belong into the standard.

Proposed Improvement:

Add

6.1.5.8 Trader storage policy

A policy of the trader that logically structures the offer space.

Note: may also include statements about damage control.

Item: 20
Clause: New
Importance: Technical
Rationale:

See item 16.

Proposed improvement:

Add

6.1.5.8 Import request acceptance policy

A policy of the trader that restricts the set of accepted import actions.

Item: 21
Clause: New
Importance: Technical
Rationale:

See item 16.

Proposed improvement:

Add chapters:
6.1.5.10 Trader assignment role
6.1.5.11 Global search policy
6.1.6.12 Resource consumption policy
6.1.6.13 Service type federation policy
6.1.6.14 Trader management security policy.

6.1.5.10 Trader assignment role

Trader assignment role regulate the arbitration between policies of clients,servers and traders, whenever these policies contradict.

6.1.5.11 Global search policy

Global search policy allows to guide the trader to import service offers from other traders. Global search policy is related to the known graph of traders.

6.1.6.12 Resource consumption policy

Resource consumption policy restricts the amount of resources used for an individual service request.

6.1.6.13 Service type federation policy

Service type federation policy consists of a reference to a set of known target traders, each with an optional and possibly incremental list of available service types intercepted into the locally understood set of types. Also the list of interceptors needed when importing from a target trader is needed.

6.1.6.14 Trader management security policy

Trader management security policy restricts the set of objects allowed to change trader properties.

Item: 22
Clause: 6.1.5.6
Importance: Technical
Rationale:

Local search policy defines within one trader the strategies of searching suitable offers. Global search policy includes the analogous strategies to be followed when spreading the search across several traders.

Proposed Improvement:

Replace 6.1.5.6 by:

6.1.5.6 Local search policy

Local search policy is tightly related to the storage policy of the trader. Local search policy allows to guide the search of suitable service offers through the logical storage structure in a optimal way.

Item: 23
Clause: 6.2.1
Importance: Conceptual
Rationale:

Introduction to the enterprise viewpoint of federation of traders should come in in this chapter. We consider federation to be a temporary group of traders, formed during an service importing activity that spans to several traders.

Some exporting is needed beforehand to configure the global community of traders to know each other, but this we consider not to be under the term "federating".

Joining into the federation of traders happens implicitly: when a trader accepts an import operation from another trader, the acceptance is taken as an announcement of joining the community. Similarly, when the original client receives the offers, the federation of traders terminate.

This view to federation must be split between viewpoints. The enterprise viewpoint talks about joining and leaving a federation. The information viewpoint shows what data is needed for the federation to succeed. The computational viewpoint shows, that no extra operations for establishing or terminating a liaison called federation is needed.

Proposed Improvement:

Continue 6.2.1. by:

A service importing activity may span several trading communities. For this kind of activity the trading communities join temporarily into a federation.

Item: 24
Clause: 6.2.3
Importance: Technical, editorial
Rationale:

Syndicates.

Proposed Improvement:

Replace text by:

A trader provides a trading function under the control of trader policy. In trading federation:

- a trader is not obligated to perform actions for another trader
- each trader is autonomous and each trader owner has complete control over the global search policy and service type federation policy.

Item: 25
Clause: 6.2.4., line 1
Importance: Editorial
Rationale:

Plural needed.

Proposed Improvement:

Replace service by services.

Item: 26
Clause: 6.2.4, last two lines
Importance: Technical
Rationale:

Statement about a compulsory expire date in service offers is too strict. It allows only one technology to be used. It also contradicts with 6.1.4.1.

Proposed Improvement:

Replace the sentence by:

Trader community may require, that the trader has a mechanism to timely remove stale service offers.

Note:

Example 1: Trader service offer acceptance policy may demand, that each offer is provided with an expire date. Trader is then allowed to remove expired offers by trader storage policy (trader is not to reveal expired offers).

Example 2: Trader import action policy may restrict revealable offers so that only those servers reachable at selection time are shown.

Item: 27
Clause: 6.2.5
Importance: Conceptual
Rationale:

If exported offers have by nature very tiny time to live, it is not reasonable to make backups. Probably middleware applications of trading would benefit, but other applications may suffer. Backups should also be included into the trader storage policy.

See also item 19.

Proposed Improvement:

Remove the line "exercise damage control (e. g. backups)."

Item: 28
Clause: 7.1
Importance: Editorial
Rationale:

Links and syndicates are being "fold in" into the properties of "normal" trader.

Proposed Improvement:

Remove last two bullets from 7.1. Overview.

Item: 29
Clause: 7.2.1.1
Importance: Editorial
Rationale:

Cyclic definition.

Proposed Improvement:

Replace last word interface by viewpoint.

Item: 30
Clause: 7.2.2.1
Importance: Editorial
Rationale:
Proposed Improvement:

Replace last word trader by trading community.

Item: 31
Clause: 7.2
Importance: Technical
Rationale:

Cleaning out syndicates and links. Links are replaced by trader properties, i.e. federation query policy, and link properties by trader property values.

Proposed Improvement:

See following items.

Item: 32
Clause: 7.2.2.2, 7.2.2.5, 7.2.2.9, text between 7.2.2.10 and 7.2.3.
Importance: Technical
Rationale:

See 31.

Proposed Improvement:

Remove link property and link property type.

Item: 33
Clause: 7.2.2.6
Importance: Editorial
Rationale:
Proposed Improvement:

Add word service: Trader property type – a subtype of service property type ...

Item: 34
Clause: Unnumbered between 7.2.2.10 and 7.2.3
Importance: Editorial
Rationale:

Discontinuity of text.

Proposed Improvement:

Add title "7.2.2.11 Z descriptions of property types" after the two first lines of 7.2.2.10.

Item: 35
Clause: 7.2.4.2
Importance: Editorial
Rationale:
Proposed Improvement:

Replace ".. unique to a trader" by " .. unique to a trading community".

Replace ".. and accessible only to .." by ".. revealed only to...".

Item: 36
Clause: 7.2.6
Importance: Editorial
Rationale:

Connection between service requests (imports) and criteria is missing.

Proposed Improvement:

Replace title Criteria by Service request.

Replace the first line in 7.2.6 by:

Service request describes an requested service. It consists of matching criteria,selection criteria, and search constraints.

Item: 37
Clause: 7.2.6.2, 7.2.6.3
Importance: Technical
Rationale:

For some applications of trading function, selection that gives only one offer is not useful. Instead a selection operation that can be parameterized by the accepted and demanded number of offers would satisfy both types of applications.

Proposed Improvement:

Replace 7.2.6.2 by:

7.2.6.2. Selection criteria – rules which are applied to the set of service offers to yield the requested number of offers.

Change in last sentence of 7.2.6.3 "a single service offer" to "required number of service offers".

Item: 38
Clause: 7.2.6.3
Importance: Technical
Rationale:

In federation, search constraints are passed from trader to trader. Each trader applies them to the local offer pool and following its import action policy passes the search on to other traders. Search constraints do not guide the search through the trader network, but the search constraint may include a import policy statement restricting the federation activity.

Proposed Improvement:

Remove the sentence "When evaluated at a particular starting trader ...".

Item: 39
Clause: 7.3.1.1, 7.3.1.3, top of page 13
Importance: Technical
Rationale:

Cleaning up the links.

Proposed Improvement:

Replace last sentence of 7.3.1.1 by:

It consists of a trader offer domain and a set of trader properties. Trader properties include knowledge about liaisons with other traders.

Remove 7.3.1.3 (two lines on page 12).

Between 7.3.1.3 and 7.4., at the top of page 13:

- Replace references to three component by references to two components.
- Remove the sentences defining trader_link_domain both from text and the Z specification.

Item: 40
Clause: 7.4.1
Importance: Technical
Questions:

- Does the trader have a set of service offers when it is first born? Hopefully the answer is NO.
- Is the initial state a product of instantiation or also a product of reactivation? No easy answers here. We refer to the discussion on Part 3. Is this resolved in the general case?

Item: 41
Clause: 7.5
Importance: Editorial
Rationale:

Dynamic schema should discuss changes not behaviors.

Proposed Improvement:

Replace the first sentence:

This sub–section presents schema which describe the interactions changing the state of the trader information object.

Item: 42
Clause: 7.4., 7.5.and 7.5.6, 7.5.7
Importance: Technical
Rationale:

Following the ODP information language structuring rules.

Proposed Improvement:

Move Search and Select descriptions from 7.5. Dynamic Schema to 7.6. Static Schema. Respectively, move Search and Select descriptions in 7.5.6 and 7.5.7 under 7.4.

Item: 43
Clause: 7.5
Importance: Technical
Rationale:

Links are now presented as trader properties.

Proposed Improvement:

Replace interactions add link and remove link by:

Modify trader properties – change the trader property values. Property values may include statements about trader policies, technology, engineering, information and enterprise aspects of trading service.

Item: 44
Clause: 7.5.1, 7.5.2, 7.5.3, 7.5.4
Importance: Editorial
Rationale:

Change in Part 3 information viewpoint language.

Proposed Improvement:

Remove sentence "In terms of the ODP information language it is a (de/re)classification."

Item: 45
Clause: 7.5.4, 7.5.5
Importance: Technical
Rationale:

Links are now presented as trader properties and these interactions are replaced by "modify trader properties" interaction.

This interaction allows also the modification of trader policies.

Proposed Improvement:

Replace 7.5.4 and 7.5.5 by:

7.5.4 Modify trader properties

The dynamic schema "Modify trader properties" defines the behavior associated with adding, deleting or modifying service properties of a trading service. These properties include trader policies and information about the owner and enterprise purpose of the trader, and aspects about contracts the trader is willing to establish (enterprise, information, engineering and technology viewpoint oriented).

Especially, for federation purposes, trader policies include service type federation policy. It states with which traders the trader under consideration is cooperating.

Note: the administrator may also force the service type federation policy to store information about the available service types through every cooperating trader. This makes it possible to utilize the technique based on federation export and import contracts, suggested in the Federation Issue, resolution alternative 1.

Item: 46
Clause: New
Importance: Technical
Rationale:

Other traders need some information about the properties of a trader allowing federation. For this, an interaction must be offered.

The trader is now handled as it were a normal service provider that may have an export policy controller and manager (administrator) of its own. The trader itself may play both the server and policy controller roles.

The trader need not necessarily reveal all its properties to other traders when it allows interactions in regard to getting information about the trader itself.

This interaction does not affect the state of the trader object, thus it should be described in chapter of static schemas.

Proposed Improvement:

Add

7.4.3. List trader properties

The static schema List trader properties is the behavior that returns the service property values of the trader to the requesting object.

Output of this interaction can be considered as a service offer. Inputs to the interaction identify the requesting object so that the trader may apply its security and exporter policies.

Item: 47
Clause: 7.5.4, 7.4.3 (new)
Importance: Editorial
Rationale:

Z specifications for the interactions described in items 45 and 46.

Proposal:

Include a Z specification.

Item: 48
Clause: 7.4.1
Importance: Technical
Rationale:

Link domain is described as trader properties.

Proposed Improvement:

Remove from Z specification the two lines containing references to links.

Item: 49
Clause: 7.5.6
Importance: Technical
Rationale:

Searching and selection interactions should obey the policies forced on the trader.

Proposed Improvement:

Add into the chapter under Z specification:

Furthermore, the search_result is intersected by sets obtained by applying trader policy rules.

Replace the line:
search result!= matching criteria? Λ
by:
search result!= matching criteria? Λ
trader_security_policy Λ

Item: 50
Clause: 7.5.7
Importance: Technical
Rationale:

Searching and selection interactions should obey the policies forced on the trader.

Proposed Improvement:

Replace in Z specification the line: selection!= selection_criteria?
by: selection!= selection_criteria ? Λ trader_security_policy

Item: 51
Clause: 7.5.6
Importance: Technical
Rationale:

Searching and selecting should involve federation aspects.

Proposed Improvement:

Add after the first chapter under Z specification:

Search may span several autonomous trader domains. This behavior is restricted by importer policy and global search policy. The trader produces a set of target trader_information_objects and invokes a search interaction on those objects (recursively?). These new search interactions take as input a intersection (or union?) of original importer policy, trader_security_policy, and service_type_federation_policy. Those target traders return search_results and those sets are combined at the requesting trader to a search_result for the importer.

Add corresponding details into the Z specification.

Item: 52
Clause:
Importance: Conceptual
Rationale:

The trader has responsibilities in achieving transparencies in distributed systems. Therefore, it should hide possible failures in achieving resources or processing. It should not itself refuse from giving out available information, if a task can not be totally finished. The trader should help to select an available server for a client to perform a task. If some other servers have failed, and their properties can thus not be known, although their offers are still in the trader repository, this should not hinder the client activity.

Moreover, the trader can have an import action policy restricting there sources used for a request. This policy can stop the processing within the trader, before a complete set of service offers have been gathered. This should not be a fatal failure. The client should be provided with the result reached so far and only warned about the unorderly stopping condition.

The results of search and select actions described in the information viewpoint should cover also these possibilities. The sets of offers defined earlier are only maximal sets, and the implementation can undeterministically produce a partial result.

Proposed Improvement:

Results of search and select actions should be restricted by import action policy (resource consumption policy). Is there a possibility to model this in Z language?

Item: 53
Clause: 7.5.7
Importance: Technical
Rationale:

Searching and selecting should involve federation aspects.

Proposed Improvement:

Make the analog modifications to select as have been done in item 51 to search.

Item: 54
Clause: 8.0, Editors question
Importance: Conceptual
Answer:

Grouping all interactions into two computational interfaces follows neatly the ODP reference model. It does not restrict the engineering description. The suggested model of two objects, one providing operations for exporters and another providing operations for importers can be drawn from the same computational interface.

Item: 55
Clause: 8.1.1
Importance: Editorial
Rationale:

Earlier in the text more restrictive description about the interface identifier type was given. To make it easier to other standards to specify this basic type in a utilizable way, all assumptions made in trader standard should be collected here.

Proposed Improvement:

Add, at least, following:

Using interface identifier value the location of the interface can be found.

Item: 56
Clause: 8.1.7
Importance: Editorial
Rationale:

Trader name type is not used.

If trader names are built by concatenating names, a restriction on allowed naming system is done. This allows only "UNIX–style" pathnames. This seems to be source routing. Is it really wanted?

Proposed Improvement:

Remove clause or replace by:

It identifies a trader object. It identifies the target trader using a local naming system method and situates the target in the global graph of traders relational to the naming trader itself.

Item: 57
Clause: 8.1.12
Importance: Editorial
Rationale:

Technology decision taken in computational viewpoint.

Proposed Improvement:

Remove word Integer.

Item: 58
Clause: 8.1.12
Importance: Technical
Rationale:

Editor request.

Proposed Improvement:

Add specifications for possible error codes as follows:

Operations described by chapters 8.2 and Annex E are interrogations by nature. Each operation has a termination alternative called error. In this termination action, an error code is returned to the interaction invoker. The following list specifies the cases in which the interaction invoker must get a termination action:

a) Unable to fulfill the given task with allowed resources.
b) Unable to fulfill the given task due to policy restrictions.
c) Unable to fulfill the given task due to lack of service offers.
d) Unable to fulfill the given task, because the task description was not fully understood: either service type or selection/search criteria could not be interpreted
e) Infrastructure error in trader domain.

Item: 59
Clause: Annex C.3, 8.1.12
Importance: Editorial
Rationale:

Should the suggestions in annex C.3 be accepted as an integral part of the standard?

C.3 suggests following error codes:
SERVICE_TYPE NOT REGISTERED
INVALID RETURN SERVICE PROPERTY NAMES
INVALID MATCHING SERVICE PROPERTY NAMES
INVALID MATCHING CONSTRAINT
INVALID SELECT CRITERIA
SERVICE TYPE NOT ALLOWED
SERVICE OFFER ALREADY EXPORTED
NO SERVICE PROPERTY VALUES
INVALID SERVICE PROPERTIES
SERVICE TYPE CONSTRAINT FAILURE
SERVICE OFFER NOT EXPORTED
MODIFIED SERVICE PROPERTIES NOT EXPORTED
STATIC MODIFIED SERVICE PROPERTIES
ADD SERVICE PROPERTIES ALREADY EXPORTED
INVALID ADDED SERVICE PROPERTIES
DELETED SERVICE PROPERTIES NOT EXPORTED
INVALID DETECTED SERVICE PROPERTIES

Some of these are not totally accepted. Is it necessarily an error to export an offer already exported? Does every server need to have values for all properties defined in service type? Is it necessarily an error to withdraw an offer already withdrawn? Operations make no harm when repeated, they are idempotent.

Chapter 8.1.12 discusses types in computational viewpoint. Annex C.3 seems to have a engineering viewpoint language in use. Should the annex be in computational viewpoint?

Proposed Improvement:

Remove from annex C.3 error codes
SERVICE OFFER ALREADY EXPORTED
NO SERVICE PROPERTY VALUES
SERVICE OFFER NOT EXPORTED
STATIC MODIFIED SERVICE PROPERTIES
DELETED SERVICE PROPERTIES NOT EXPORTED.

Item: 60
Clause: 8.1.12
Importance: Conceptual
Rationale:

Interrogations have several alternative termination behaviors. So far, 8.1.12 implies two possible termination behaviors for all trader operations. These are "normal reply from trader" and "error code from trader". There should also be a third kind of termination, namely the case that the trader is not reached or it fails. In this case, the environment in which the client and trader interact, produce an infrastructure error and it can be passed to the client as a termination action within the interaction.

Proposed Improvement:

Add in 8.1.12 a set of error codes for infrastructure errors:

- target of interrogation not reached
- no reply for interrogation
- failure of interrogation target
- failure of communication channel
- failure of client infrastructure.

Item: 61
Clause: 8.2, Annex E
Importance: Conceptual, editorial
Rationale:

See item 60.

Proposed Improvement:

Change note on 8.2 line 2 to read:

Note: at operation signature tables head is the initiation action and the parameters are classified as M=mandatory and O=optional, and termination shows the three possible termination actions, S=success, TF=trader failure, IF= infrastructure failure.

Replace in each table the F's with "TF or IF".

Item: 62
Clause: 8.2, Annex E
Importance: Editorial
Rationale:

In replies that the trader provides there often is a list of items. Each item consists of a identifier, related property values, etc. In signature parameters these are currently expressed as a list of identifiers, a list of property types and so on. The grouping of related items together is missing.

Proposed Improvement:

Add an explaining line into the tables: for example in search

"list of items consisting of

1)	interface identifiers	interface identifier
2)	service property values	property value list type
3)	service offer property values	property value list type".

Item: 63
Clause: 8.2, Annex E
Importance: Editorial
Rationale:
Proposed Improvement:

Express explicitly in behavior descriptions that operations are interrogations.

Item: 64
Clause: 8.2, Annex E
Importance: Editorial
Rationale:

Placement policy of the trader contradicts the autonomy of traders. Other mechanisms for placement of offers and pointing suitable trader graph for search can be used. Search graph may be restricted by importer policy or search constraint.

In the computational behavior of operations a sentence "If the invocation parameters specify a trader name then the search operation is invoked on interface determined by the first component of the name, with the remaining components substituted as the trader name." may be interpreted several ways.

Proposed Improvement:

Remove the above mentioned sentence from behavior descriptions.

Item: 65
Clause: 8.2.1
Importance: Editorial, conceptual
Rationale:
Proposed Improvement:

Lines 18−19 at page 23, replace " a linked trader" by "a trader in liaison".

Lines 11, 17 at page 23, replace "search policy " by "search parameters".

Line 11 at page 23, replace "trader search policy" by global search policy.

Line 10 at page 23, remove words "expressed as a"

Line 19 at page 23, add ". Global search policy states whether the search is initiated at federated traders serially or parallelly."

Item: 66
Clause: 8.2.2
Importance: Technical
Rationale:

Trader should give as much service as it is possible, even if the requested task can not be completed.

It should for example give all the property values it recognizes, although it would not know all requested properties.

Proposed Improvement:

Remove line "Service properties listed are not defined for this service type." from the list of total failures in the bottom of page 23.

Item: 67
Clause: 8.2.4
Importance: Editorial
Rationale:

Earlier items on placement policies.

Proposed Improvement:

Replace computational behavior by:

Depending on the trader security policy and the trader service offer acceptance policy, the offer is either discarded or added into the traders offer domain. An identifier (unique with respect to that trader) is generated, stored and passed to the invoker. The operation is an interrogation.

Item: 68
Clause: 8.2.4, 8.2.6
Importance: Technical question
Rationale:

Is there restrictions on the property values that the export operation should tell about the offer? Is there a division to mandatory and optional properties? If there is no such division, are all properties optional or mandatory? In all three cases, some corrections must be made through the text. Currently some parts (for example error codes) suggest that all properties are mandatory, while some other parts (for example operations like replace operation) suggest that all properties are optional.

Item: 69
Clause: 8.2.6
Importance: Technical
Rationale:

Trader is ABLE to handle virtual property values (= is able to request values from an external source, i.e. exporter policy controller). Do we still need a MANDATORY operation to change property values?

Proposed Improvement:

Move 8.2.6 Replace operation to Annex E Non Normative Computational Specification of the Trading Function.

Item: 70
Clause: Annex E
Importance: Editorial
Rationale:

Avoiding misunderstandings.

Proposed Improvement:

Change the name of Annex E to: Normative Specification of Optional Computational Interactions of the Trading Function.

Item: 71
Clause: 8.2.5
Importance: Editorial, technical
Rationale:

What is the decision based on when the authorization of client is checked? All allowed exporters may not withdraw all offers. The authority can neither be based only to the holding of a offer id.

Proposed Improvement:

Replace in computational behavior "decision given by any security service regarding the client id" by "decision given by the security service(s) used by the trader regarding the client id and offer id".

Item: 72
Clause: 8.2.7, 8.2, Annex E
Importance: Technical, editorial
Rationale:

Editor's request in 8.2.7.

Proposed Improvement:

Remove 8.2.7.

Add to each operation description in chapter 8.2 and Annex E rules that tell in which situations the operation is available: "Operation is available to a client when the client and the trader are in liaison."

Item: 73
Clause: 8.3, Annex E
Importance: Conceptual
Rationale:

Links are represented as trader properties.

Combine operations Add link, Remove link, Modify link, Add trader policy, Delete trader policy, Modify trader policy into one operation called Modify trader property.

Proposed Improvement:

Remove 8.3.1, 8.3.2, 8.3.3, E.2.2, E.2.3 Delete and E.2.3 Modify.

Add:

8.3.1 Modify trader properties

Function

Modify trader properties is an interrogation that allows the administrator to set the policies of the trader.

The properties set by the administrator introduces

– policies to be followed when accepting service offers and service requests from clients
– contract templates to be used when other traders wish to establish a binding for federation purposes.

Signature

PARAMETER	PARAMETER TYPE	HEAD	TERMINATION
client id	identifier type	M	
list of items consisting of		M	
1) trader property name	property name type		
2) trader property value	property value type		
			S
error code	error code type		TF or IF

PARAMETER	DESCRIPTION
client id	Identifies the object that invoked the operation.
trader property name	Each trader policy has a corresponding property name.
trader property value	The effective policy rules.

Specially service type federation policy includes names and interface identifiers for the cooperative traders. The property value must also include knowledge about interceptors needed when establishing a binding for the federated import. This property value may also be supplemented with federation export and import contracts, if the administrator decides so.

Also important for federation is that the used type management system is identified as a property of the trader.

Computational behavior:

Depending on the decision given by a security service regarding the client id and the trader administrator id known by the security service from earlier interactions, the operation is either rejected or accepted. The trader property name selects the policy on which the modification, i. e. addition or removal or an atomic combination of both removal and addition, is done.

Item: 74
Clause: 8.3.4
Importance: Conceptual
Rationale:

Continuation for change started in 73.

Proposed Improvement:

Replace 8.3.4 by:

8.3.4 List trader properties

Function

List trader properties allow the administrator to view the state of the trader properties. It also allows other traders or other objects to view trader properties in order to decide if the trader is suitable for a federation.

Signature

PARAMETERS	PARAMETER TYPE	HEAD	TERMINATION
client id	identifier type	M	
property names	property name type list	M	
trader identifier	identifier type		S
list of items consisting of:			S
1) property name	property name type		
2) property value	property value type		
error	error code type		TF or IF

PARAMETER	DESCRIPTION
client id	Identifies the object that invoked the operation.
property names	The names of the properties whose values should be returned.
trader identifier	Identifies the responding trader object.
property value	Values retrieved from the trader policies.

Computational behavior:

Operation is an interrogation. The response is created so that the security policy restricts the revealed properties.

Item: 75
Clause: 8.3.5
Importance: Editorial, conceptual
Rationale:

Editor's question.

Proposed Improvement:

Act as in item 72.

Item: 76
Clause: E.1.1
Importance: Technical
Rationale:

Resolve contradicts the autonomy of trading domains.

Proposed Improvement:

Remove E.1.1.

Item: 77
Clause: E.2.1
Importance: Editorial
Rationale:

Operation not needed.

Proposed Improvement:

Remove E.2.1.

Item: 78
Clause: E.3.2
Importance: Editorial
Rationale:

Exporter is a role of a logical object within the trading community. A logical object may include properties, policies, management operations and mission specific operations. It may also include operations for querying policy decisions of the object. The logical object may be composed of several objects.

Proposed Improvement:

Change name of exporter policy controller to Exporter.

Item: 79
Clause: E.3.2
Importance: Conceptual
Rationale:

This operation could be removed if also the exporter policy were allowed to be viewed as a virtual property value that is evaluated by the trader by demand using status enquiry.

Two possible scenarios exists on checking that the exporter allows an offer to be revealed to the client:

– dynamic property: trader requests exporter for allowance

– static property: trader knows the policy of exporter and applies it on clients and offers.

Proposed Improvement:

Remove E.3.2.

Item: 80
Clause: A.1, line 8
Importance: Conceptual
Rationale:

Trader cannot be forced into one viewpoint only, at least not without telling in which system specification this viewpoint is taken. Also the tutorial should talk about the trading FUNCTION. Small editorial modifications are suggested at the same time.

Proposed Improvements:

Replace lines 8–20 of paragraph A.1 with:

Open Distributed Processing aims to provide distribution transparent utilization of services over heterogeneous architecture software and hardware platforms, and heterogeneous networking environment. To advance the interworking of software components in this heterogeneous environment an ODP common function, trading function, has been introduced.

In order to utilize services in the network, service users need to be aware of the potential service providers and capable of accessing them. Furthermore, as in large distributed systems, group of sites and applications are frequently changing, it is advantageous to allow late binding between service consumers and service providers. When late binding with trading function is supported, a software component is always able to utilize an appropriate service provider, although the requirements of the component change dynamically and the state of the network and its services change dynamically.

Item: 81
Clause: A.2, line 1
Importance: Conceptual
Rationale:

Trader does not form the binding between client and server. Linking is not any more an ODP term.

Proposed Improvements:

A trader is a third party object that enables the clients to find suitable servers in a distributed system.

Item: 82
Clause: A.2, second bullet,line 2
Importance: Editorial
Rationale:

Trader doesn't any more bind the clients to servers.

Proposed Improvements:

Change " ... a client requires a service." by " ... a client requires knowledge about a service."

Item: 83
Clause: A.2.1
Importance: Editorial
Rationale:

Text gives only telecommunication oriented examples. Other areas of applicability would also deserve a few words. We propose to include into this description the sets or categories of services mentioned by Mirion Bearman in her message from 17.5. 1994:

" Many (overlapping) schemes are possible, for example:

* categories based on existing Internet resources, e.g. research papers, library catalogs, internet shopping, archive services;
* industry−based categories for further decomposition by an appropriate industry body, e.g. telecommunications industry, software industry, tourist industry;
* country−based categories, for further decomposition by standards organizations within each country.

There is no reason why these categorisations cannot be used concurrently, giving maximum flexibility and some redundancy (leading to reliability). Trading policy of the local trader could prioritise particular categorization schemes based on the needs of the local organization. Organizations are also free to choose which" *traders* " they wish to use, both for advertising their own services and importing other services."

However, we do not agree on standardizing the categories allowed within the Trading function standard! Those standards should be separate and the traders should only use external sources of service type information.

Item: 84
Clause: A.3.2.1
Importance: Conceptual
Rationale:

Most ODP oriented projects seem to keep the trading function as a middleware component. Trading suits in that role perfectly. It only needs a suitable type manager with knowledge about middleware services. If the type manager contents is not restricted only to middleware services, the usability area of trading grows.

Trading function can be utilized also in several other ways, in several viewpoints. Trader may serve for example as "an enterprise repository" as have been done in [O'Connor, S., Describing Systems in the OPENframework Integration Knowledge Base. ICL Technical Journal, Volume 8 Issue 3, May 1993.]

Trading function and a set of standardized service types needed for an application area should together form a tool for system builders. We seem to need 1) trading function standard, 2) set of standardized service type groups used via traders to form standard system platform, standard application level platforms and so on.

Item: 85
Clause: A.3.2.4
Importance: Editorial
Rationale:

Context were some time ago an important feature in trading function. However, the internal storage structure of a trader can not be prescribed in trading function standard. Trader storage policy may force the storage structure used within a trader. The policy may be formed by organizational structure, by service types, or be based on grouping rules.

For the internal offer groups, context is not a very good name. ODP reference model uses the term for a contractual state of an objects. Thus, trader context may mislead the discussion.

The discussion within A.3.2.4 is good work and could be used in a different position of trading function standard. There is several other policies that could have similar discussion about various possibilities to help system designers. These descriptions could be gathered together into a "guide to use" annex.

As we can not provide the same level of discussion with other policies before Southampton meeting, we should balance the text for going to CD by removing A.3.2.4 for the time being and trust that this piece of work is brought back at appropriate time when annex "guide to use" is constructed. This annex can be added at CD stage.

Proposed Improvements:

Remove A.3.2.4.

Add title for Annex x Guide to Use.

Item: 86
Clause: A.3.3.1, page 45, lines 20–24
Importance: Editorial
Rationale:

Representing behavior by name value pairs seems to confuse readers.

Proposed Improvements:

Replace Note by:

Note: Behavior descriptions can be stored as name value pairs. Name is the label of the action described and the value is the action description. Also the environmental constraints can be presented this way: pairs labeled with names QoS–requirements, deadlines, availability,security, fault tolerance and location can contain restrictive rules as corresponding values.

Item: 87
Clause: A.3.3.1, page 45–46
Importance: Editorial
Rationale:

Changes in the corresponding prescriptive text.

Proposed Improvements:

Remove from page 45 last 3 lines and from page 46 first 5 lines.

Item: 88
Clause: A.3.3.1, page 46, line 16–17
Importance: Technical
Rationale:

This is only one decision that the trader assignment policy can force.

Proposed Improvements:

Use the sentence "The trader policy overrides client policies when conflict exists between the two policies." as an example of trader assignment policy.

Item: 89
Clause: A.3.3.3
Importance: Editorial
Rationale:

Changes in the corresponding prescriptive text.

Proposed Improvements:

Remove bullet Export Selection.

Item: 90
Clause: A.5
Importance: Editorial
Rationale:

Syndicates removed in earlier comments.

Proposed Improvements:

Remove chapter A.5

Item: 91
Clause: A.6
Importance: Editorial
Rationale:

Syndicates removed in earlier comments.

Proposed Improvements:

Change the title of A.6 to Federated trading

Remove titles A.6.1. and A.6.2

Item: 92
Clause: A.8, A.2.1
Importance: Standardization technique
Rationale:

Service Interaction Manager is an important application for trading function. Now the title is hidden under a tutorial annex. To this application more attention should be drawn.

Proposed Improvements:

Remove A.8.

Add into A.2.1 a paragraph referencing to SIM as a user of trading function.

Take care that SIM is kept on the list of needed "receipt standards".

Include contributions on SIM into Part 1.

Item: 93
Clause: H.3.1, page 90
Importance: Editorial
Rationale:

The second figure is uninformative.

Proposed Improvements:

Remove the second figure.

Item: 94
Clause: H.4.1
Importance: Technical
Rationale:

Trader can use X.500 directory as a external server. Clients use trader as an tailorable user interface to the Directory. Trader may use a service type for the Directory that gives only a partial view to the information stored into the Directory.

Proposed Improvement:

These aspects should be taken into account in the text.

Item: 95
Clause: H.4.1
Importance: Technical
Rationale:

Traders can use X.500 Directory as an external server that knows about locations of other traders. Each trader may search other traders from the directory as largely it wishes. It may also ignore the X.500 and request information about available trading services from other traders via federation established by the administrator with "out of the system" information. To use the Directory as a database of traders requires that administrators export their traders into the Directory or at least allow other traders to use "list trader properties" operation.

Proposed Improvement:

These aspects should be taken into account in the text.

Item: 96
Clause: H.4
Importance: Technical
Rationale:

X.500 can be used as the storage function of trader. Technically service offers include only static information: static values and static references to interfaces from which dynamic values can be retrieved.

Proposed Improvement:

These aspects should be taken into account in the text.

Item: 97
Clause: H.2, page 89
Importance: Editorial
Rationale:

Wrong structure in the bullet list in third paragraph on page 89.

Proposed Improvement:

Tabulate the third and fourth bullet under the sentence "trader triggers ...". Both bullets should start by word "while".

Item: 98
Clause: Annex H
Importance: Conceptual
Rationale:

For us it is unclear that the trading function standard should prescribe how the trader storages its data.

Proposal:

We suggest that the trader group should request clarification from WG7 on this matter.

Item: 99
Clause: Annex F
Importance: Technical
Rationale:

Within the trading function standard engineering aspects can not be normatively specified. However, to apply trading, such norms must be available. Engineering viewpoint restrictions deal with policy rules. To make federation usable in an application, for example SIM, there must be common semantics for the policies and of course for the service types,so that cooperation can be reached.

In the item 85 we suggested that a new annex "guide to use" would be made. This annex with annex F should provide enough material for most application area trading systems.

However, this annex will grow overwhelmingly. It will produce a whole set of "receipt standards" i.e. "procedure standards".

Proposed Improvement:

Instead of using an annex F, tell already in the Introduction that engineering specifications are done in "receipt standards", not in trading function basic standard.

Item: 100
Clause: Annex G
Importance: Technical
Rationale:

See item 99.

Technology viewpoint policies probably deal with time constraints and such things needed for communication contracts.

Proposed Improvements:

Explain in Introduction likewise with engineering specification.

Item: 101
Clause: Issues list
Importance: Conceptual
Rationale:

We think that above items have given some resolution to

- allowing importers to reach offers stored in different trading domain
- allowing each trading domain to act autonomously
- allowing administrators to guide the traders by policies
- allowing traders to find each others and request remote traders possibilities to tell about offers of different service types
- allowing trader to federate even if their policies and service types differ.

The federation activity have been changed from a three stage process to a two stage process.

The traders partially provide the ODP federation transparency function themselves. Traders can thus not utilize that transparency, but they can utilize interceptors.

If the traders are on different communication domains, trader itself cannot call interceptors to make communication possible. In order to interact with the remote trader (even only for requesting for federation establishment) requires a shared mechanism for communication. The traders must therefore either to be located in same communication domain, or they must know an interceptor for their communication needs, already before the interaction is attempted. The communication domain interceptors, as well as security objects, must be brought in to the configuration by other parties, for example the instantiation of the trader or the administrator by setting trader properties.

Proposed Improvements:

Remove the issues list.