SeLeNe — Self E-Learning Networks
IST-2001-39045

WP3 Deliverable 5

An Architectural Framework and Deployment Choices for SeLeNe

The SeLeNe Consortium

Abstract

This deliverable specifies the high-level architecture of SeLeNe at an abstract level, taking into account the outcomes of previous Workpackages and also Deliverable 3 on a service-based architecture for SeLeNe. The high-level architecture is evaluated with respect to SeLeNe's functional requirements, as identified in Deliverable 2.2. This is done by using notations from UML 2.0, listing SeLeNe's main actors and use cases, and showing the sequence of services used to enact each use case. A number of concrete deployment scenarios of this architecture are then identified and evaluated. To this end, we first introduce the reasons behind the selection of our service-based architectural framework and then evaluate three approaches to service placement with regards to their feasibility using current Grid and P2P technology. SeLeNe services address a number of specific issues, and thus we need to concretise actual deployment possibilities in order to present alternatives that satisfy the different requirements for systems which make use of these services. We conclude with an Exploitation Plan for the outcomes of SeLeNe.

January 31, 2004

The SeLeNe Project

Life-long learning and the knowledge economy have brought about the need to support a broad and diverse community of learners throughout their lifetimes. These learners are geographically distributed and highly heterogeneous in their educational backgrounds and learning needs. The number of learning resources available on the Web is continuously increasing, thus indicating the Web's enormous potential as a significant resource of educational material both for learners and instructors.

The SeLeNe Project aims to elaborate new educational metaphors and tools in order to facilitate the formation of learning communities that require world-wide discovery and assimilation of knowledge. To realise this vision, SeLeNe is relying on semantic metadata describing educational material. SeLeNe offers advanced services for the discovery, sharing, and collaborative creation of learning resources, facilitating a syndicated and personalised access to such resources. These resources may be seen as the modern equivalent of textbooks, comprising rich composition structures, 'how to read' prerequisite paths, subject indices, and detailed learning objectives.

The SeLeNe Project (IST-2001-39045) is a one-year Accompanying Measure funded by EU FP5, running from 1st November 2002 to 31st January 2003. The project falls into action line V.1.9 CPA9 of the IST 2002 Work Programme, and is contributing to the objectives of Information and Knowledge Grids by allowing access to widespread information and knowledge, with e-Learning as the test-bed application. The project is conducting a feasibility study of using Semantic Web technology for syndicating knowledge-intensive resources (such as learning objects) and for creating personalised views over such a Knowledge Grid.

Executive Summary

This deliverable is part of the SeLeNe Workpackage 3, on the architecture of Self e-Learning Networks. The objectives of Workpackage 3 are:

- To identify technologies for managing RDF descriptions of educational resources in an open, evolving environment such as a Self e-Learning Network, including issues such as distribution, replication, scalability, load and locality, and define a set of services supporting SeLeNe's functionality.
- To design the high-level system architecture of SeLeNe.
- To produce an Exploitation Plan for the outcomes of SeLeNe.

The traditional approach of storing data in database management systems (DBMS) is not suitable in a Web environment. With current DBMS technology, data are relatively homogeneous, exhibit a small degree of distribution (on relatively few network sites), remain unchanged unless explicitly updated, and are largely static (their location remains fixed). These assumptions do not hold in the semi-structured computing world of the Web, thus creating the need for new foundations for all data management aspects: modelling, storage, and querying.

Metadata management in such an environment is crucial. To this end, we envisage that metadata (RDF/S descriptions) and services related to learning objects will be maintained in dedicated metadata stores. These metadata stores will create a backbone of knowledge and services and provide the means for browsing and searching for learning objects. The proposed technologies must exhibit a high degree of scalability, allowing new metadata stores to easily join the backbone. Deliverable 3 reports on these issues and on a Grid service framework for supporting SeLeNe's functionality.

In this deliverable the high-level architecture of SeLeNe is first specified at an abstract level, taking into account the outcomes of Workpackages 2 and 4, and of Deliverable 3 on a service-based architecture for SeLeNe.

This architecture is then evaluated with respect to SeLeNe's functional requirements, as identified in Deliverable 2.2. This is done by using notations from UML 2.0, listing SeLeNe's actors and use cases, and showing the sequence of services used to enact each use case.

A number of concrete deployment scenarios of this architecture are then identified and evaluated. More specifically, with respect to service placement alternatives, our aims are:

- To present how the proposed architecture relates to current work towards the Semantic Grid. Through the development of the service-based architecture we bring forth issues that arise in relevance to today's Grid technology when examined through our metadata oriented, self e-learning network.

- To explore a number of different service placement scenarios in order to produce alternative architectural models, each fitting different learning environments. By 'learning environment' we refer to the topology created by the actual service providers and service consumers, the degree of dynamicity, distribution and metadata storage (i.e. where the RDF descriptions are stored, registered and accessed). We also compare the alternatives presented in an attempt to assist the task of constructing a SeLeNe to manage the space of learning object descriptions.

We finally conclude with an Exploitation Plan for the outcomes of SeLeNe.

Revision Information

Revision Date	Version	Changes
October 31, 2003	0.1	First Draft Proposal
November 16, 2003	0.2	Second Draft Proposal
December 9, 2003	0.3	Third Draft
December 11, 2003	0.4	Fourth Draft
December 29, 2003	0.5	Extensions to UCY section
January 31, 2004	1.0	Final version - updated to UML 2.0

Contents

1 SeLeNe's Service-Based Architecture

SeLeNe Deliverable 3 [32] identifies the set of services necessary to provide SeLeNe's functionality. These services are illustrated in Figure 1 and are as follows:

Access: Provides an API allowing access to an RDF repository for retrieval and update of RDF/S descriptions. Allows nodes of a SeLeNe network freedom to implement the storage of RDF/S descriptions in any manner required, while providing a uniform interface for access to such descriptions.

Communication: Provides the basic communication mechanisms between SeLeNe services.

Information: Provides a means of discovering information about a SeLeNe node, such as which services it provides and which RDFS schemas it supports.

Query: Provides querying facilities over RDF/S descriptions using a structured query language such as RQL [26]. It may need to call the Syndication service for translation of queries expressed on one RDFS schema into equivalent queries expressed on another RDFS schema. It calls the Access service to retrieve descriptions.

Sign-on: Allows a new site to register with a SeLeNe network and advertise its content and services.

Locate: Similarly to the OGSI GridService [15], this service allows for the discovery of SeLeNe service providers.

Syndication: Translates between different RDFS schemas using *articulations* (mappings) between heterogeneous schemas [30]. Performs both data-to-data and query-to-query translations.

Update: Provides update facilities over RDF/S descriptions using an RDF update language such as that described in [29]. It may need to call the Syndication service for translation of updates expressed on one RDFS schema into equivalent updates expressed on another RDFS schema. It calls the Access service to insert, delete or update descriptions.

Event and Change Notification: Provides reactive functionality over RDF descriptions by supporting event-condition-action rules that allow detection, notification and propagation of changes to learning object (LO) descriptions and users' personal metadata [29]. This service calls the Query and Update services to evaluate ECA rule conditions and actions. This service is also responsible for building indexes at nodes of the SeLeNe network in order to facilitate the detection of events and the triggering of ECA rules. In doing so, it may need to call the Syndication service in order to translate between heterogeneous RDFS schemas (see [29]).

View: Allows the creation of personalised views over the LO descriptions and schemas, using a view definition language such as RVL [26].

LO Registration: Allows the registration of new LOs by providing their descriptions to the SeLeNe [30].

User Registration: Allows users to register with the system and to create a personal profile for use in personalisation [22].

Trails and Adaptation: Provides personalised search of LO descriptions through the

translation of keyword-based queries and by filtering and ranking search results according to a user profile [22]. This service can also generate trails of LOs tailored to particular users.
Collaboration: Allows communication between users and groups of users. The collaboration service may be used to mediate existing mechanisms such as Whiteboards, Message Boards and instant messaging services, as well as allowing for collaborative filtering [27] to recommend LOs to users based on the behaviour of other similar users.
Presentation: Formats information for delivery to the SeLeNe user interface. This includes the graphical visualisation of RDF/S descriptions, the display of trails and the presentation of search results.

The following section shows how these services interact to provide the complete functionality of a SeLeNe system, as described in SeLeNe Deliverable 2.2 [23] and detailed in Deliverables 4.1– 4.4 [30, 22, 26, 29]. The reader is referred to these documents for exact details of the functionality encompassed by each use case, but for reference we provide a brief summary of some of the terms used:
A SeLeNe **node** or **site** is a networked computer providing at least the minimal set of core SeLeNe services [32], that has registered to form part of a SeLeNe.
A **user** is an individual LO consumer or producer who has registered with a SeLeNe, and who can "log on" to a node in order to access SeLeNe's services (a node may have more than one user logged on at the same time).
Users all have a **profile**, which contains information about the user that is used to provide SeLeNe's advanced personalisation functionality.
A **group** is a collection of users who share a single "group profile" (much like the user profile) for the purpose of some collaborative activity.
A **trail** is a sequence of LOs, possibly with annotations, where the sequence represents an ordering for learner interaction with the LOs.
A SeLeNe's **LO information space** is the collection of LO descriptions and their schemas (in RDF and RDFS) that have been registered with the SeLeNe.
SeLeNe provides interfaces for LO discovery either by **browsing** the schema attributes, in a similar manner to a Web directory such as Yahoo![1], or by **searching** the LO descriptions using keywords, in a similar manner to a Web search engine such as AllTheWeb[2].
Users can also define **views** of the LO information space by defining their own (RDFS) schemas and populating them with elements from the registered LO descriptions; this is similar to the way that views of tables can be defined in a relational database. Once defined, these views can be browsed or searched directly by the user.

[1]http://www.yahoo.com
[2]http://www.alltheweb.com

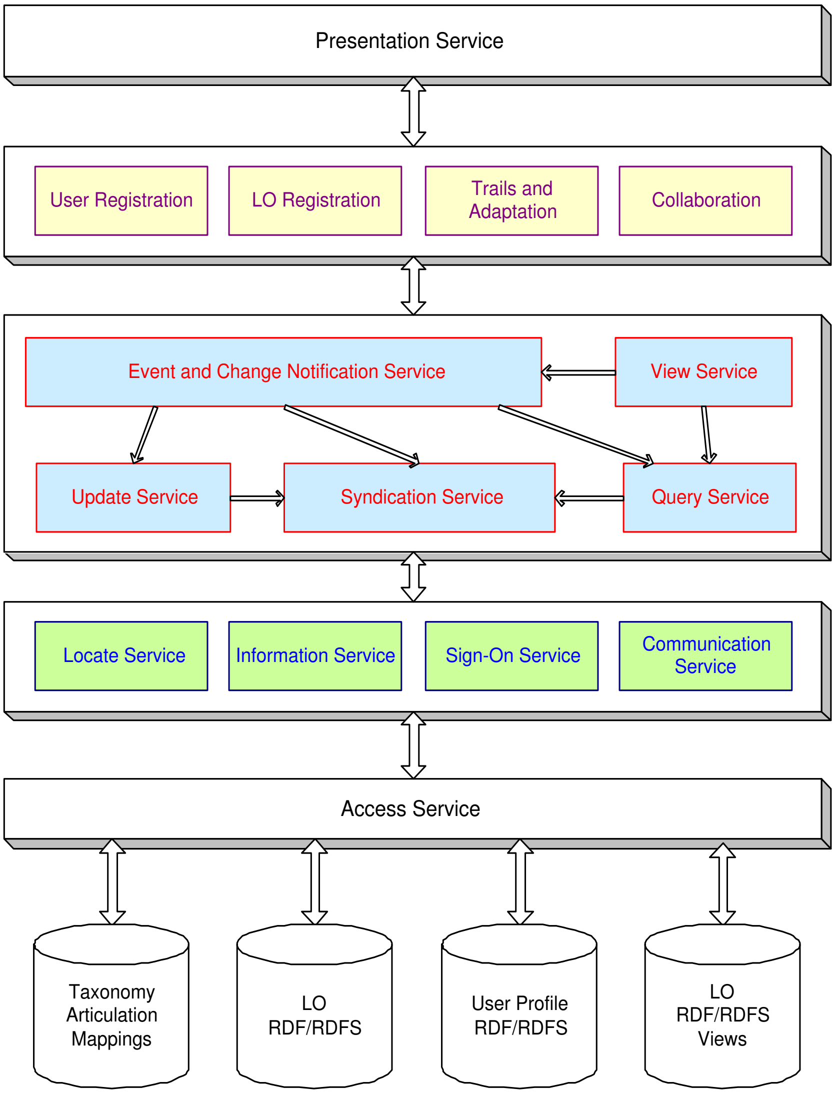

Figure 1: SeLeNe's Service-Based Architecture

2 SeLeNe Use Cases

Kevin Keenoy, George Papamarkos, Alexandra Poulovassilis
School of Computer Science and Information Systems, Birkbeck, University of London

In order to validate that the proposed service-based architecture is capable of providing SeLeNe's functionality, we here list the use cases relevant to SeLeNe (derived from *SeLeNe Deliverable 2.2: Self e-Learning Networks – Functionality, User Requirements and Exploitation Scenarios* [23]) and for each one describe the sequence of service calls that would be sufficient to enact the use case.

There are three kinds of actors identified in the following use cases:

1. Nodes participating in the SeLeNe network.
2. Human users of a SeLeNe system.
3. The SeLeNe system itself.

Where the service interactions required to enact a use case are non-trivial a UML 2.0 Sequence Diagram [18] is provided, showing SeLeNe's services as the participants.

2.1 Use Cases for Nodes

The use cases for nodes are as follows:

- a new node joins a SeLeNe;
- a node leaves a SeLeNe.

2.1.1 A New Node Joins a SeLeNe

This use case can be enacted by the following sequence of interactions:

(1) The new node contacts a known node of the SeLeNe it wishes to join and asks the *Communication* service to send a request to the *Locate* service to discover a *Sign-on* service
(2) The *Communication* service forwards the request to the *Locate* service
(3) The *Locate* service responds to the *Communication* service with the location of a *Sign-on* service
(4) The *Communication* service sends the response back to the new node with the location of a *Sign-on* service
(5) The new node sends its registration details to the *Communication* service with a request to pass them to the *Sign-on* service
(6) The *Communication* service sends the request to the *Sign-on* service
(7) The *Sign-on* service asks the *Communication* service to confirm registration with the node
(8) The *Communication* service sends confirmation to the new node.

These interactions are shown in the sequence diagram in Figure 2.

All message passing between services in SeLeNe is mediated by the *Communication* service, but as can be seen from this first simple example some clarity of explanation is lost when all *Communication* service calls are included in the sequence diagram. Therefore, **in the remainder of this document we omit the *Communication* service from the description of use case enactments**, but it should be read implicitly that all inter-service communication shown is via the *Communication* service.

Leaving the mediation of the *Communication* service implicit, the sequence diagram for this use case becomes the simpler one shown in Figure 3, and its description is reduced to:

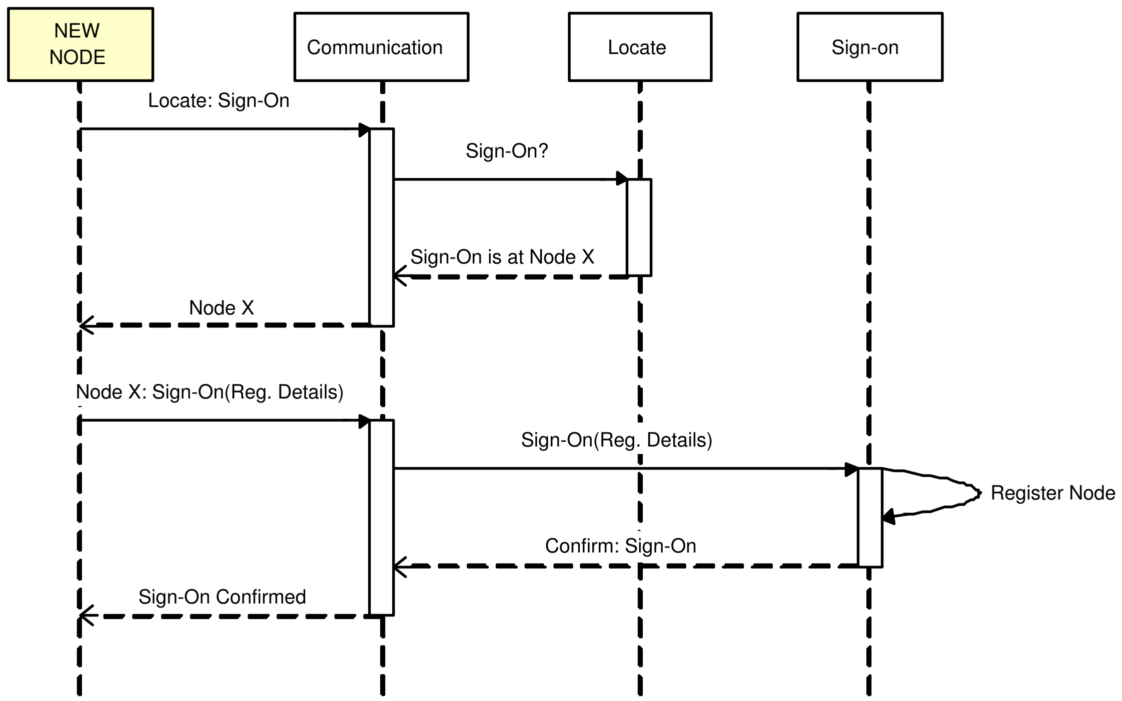

Figure 2: Node Registration

(1) The new node contacts a known node of the SeLeNe it wishes to join and asks the *Locate* service to discover a *Sign-on* service
(2) The *Locate* service responds with the location of the *Sign-on* service
(3) The new node contacts the *Sign-on* service with its registration details
(4) The *Sign-on* service confirms registration with the SeLeNe.

2.1.2 A Node Leaves a SeLeNe

(1) The node contacts the *Sign-on* Service to let it know that it is going to leave the SeLeNe
(2) *Sign-on* returns, confirming that the leaving message has been received
(3) The *Sign-on* service communicates with the *Event and Change Notification* service relevant to the node that has left, informing it of the departure of the node
(4) The *Event and Change Notification* service searches the indexes, ECA rule bases and RDFS schemas throughout the SeLeNe and removes all references to this node
(5) The *Update* service is used to remove all references to LOs, groups and users registered at this node (see Sections 2.2.14, 2.2.17 and 2.2.18 for the enactment of these removals).

The corresponding sequence diagram is shown in Figure 4. The figure shows the *Update* service calling the *Access* service to perform necessary removals. The *Update* will generally need to access the RDF descriptions to perform any updates, so **in the remainder of this document we do not show calls to the *Access* service from the *Update* service**. We leave this call implicit, as it will always be necessary.

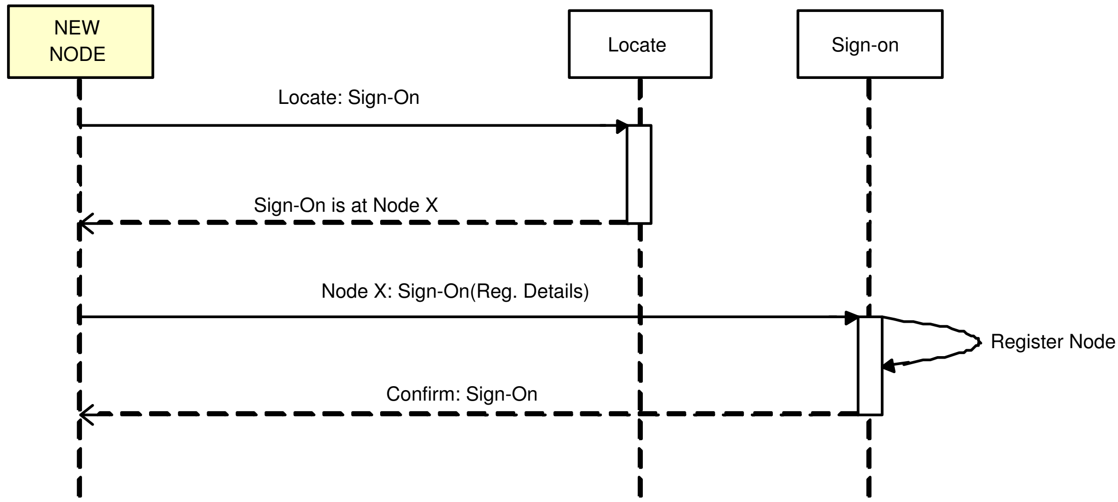

Figure 3: Node Registration, Omitting the *Communication* Service

2.2 Use cases for Human users

The use cases for human users are as follows:

- a user registers with a SeLeNe via one of its nodes;
- a user updates his/her profile;
- a user creates a new group of users;
- a user joins or leaves an existing group of users;
- a user or group registers an RDFS schema describing a taxonomy;
- a user or group registers an *Articulation* between a taxonomy they own and another SeLeNe taxonomy;
- a user or group defines a view over the SeLeNe LO descriptions;
- a user or group generates a view over the SeLeNe descriptions based on their profile;
- a user or group requests automatic notification services;
- a user registers an atomic LO with the SeLeNe;
- a user composes and registers a composite LO;
- a user or group authors and registers a trail;
- a user requests a replacement of the LOM and/or the taxonomical description of a LO;

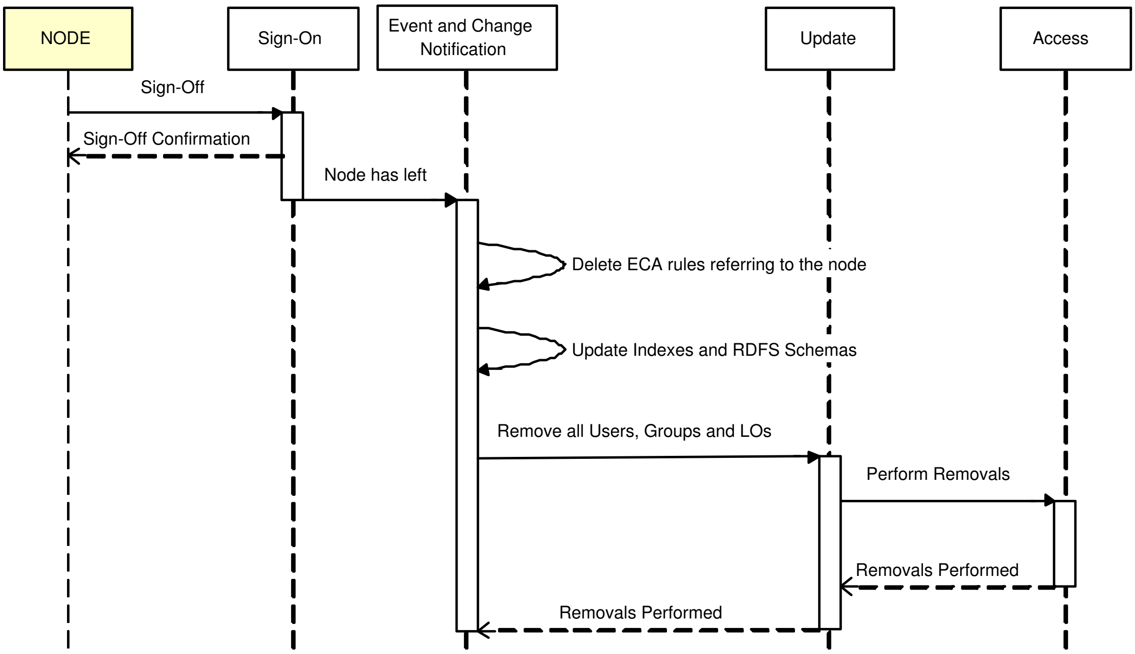

Figure 4: A Node Leaves a SeLeNe

- a user withdraws a (composite or atomic) LO which he/she owns;
- a user browses the LO information space;
- a user searches for LOs;
- a group's owner shuts down the group;
- a user leaves the SeLeNe.

2.2.1 A User Registers with a SeLeNe via one of its Nodes

(1) The user contacts a SeLeNe node and requests registration via its *Presentation* service
(2) The *Presentation* service uses the *Locate* service to discover a *User Registration* service and presents a registration GUI to the user
(3) The user provides his/her registration details to the *User Registration* service, via the registration GUI
(4) The *User Registration* service generates an RDF description of the user and this profile is then passed to the *Event and Change Notification* service
(5) The profile is sent to the *Update* service for inclusion in the local RDF repository
(6) The *Event and Change Notification* service then registers the standard ECA rules for the profile (see [22]) and then evaluates existing ECA rules that may be triggered by the addition of a new profile
(7) Registration is confirmed to the user.

This is shown in Figure 5.

The *Locate* service will need to be used to find services that need to be called by various other services in many of the enactment scenarios. As for calls to the *Communication* and *Access* services, **calls to the *Locate* service are generally omitted from the diagrams and enactment descriptions in the rest of this document**. It should be assumed that, where necessary, the *Locate* service is used for service discovery.

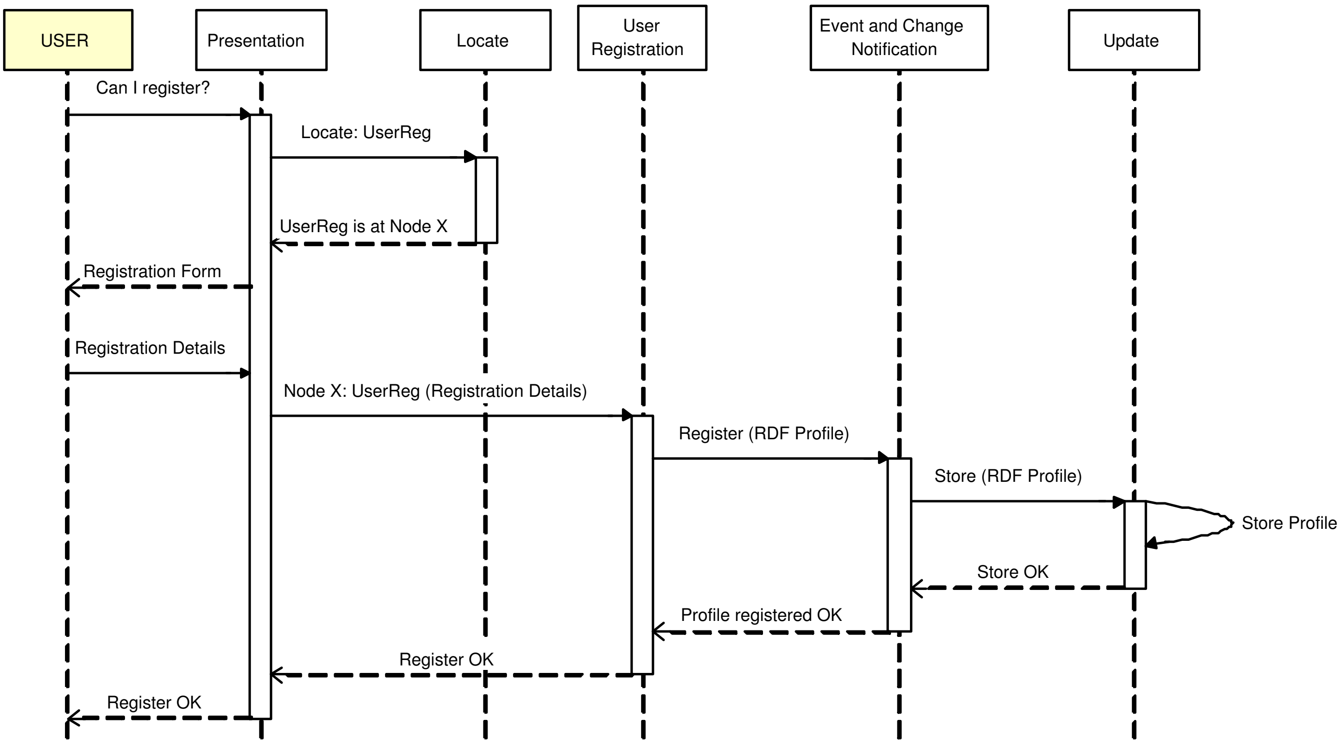

Figure 5: User Registration

2.2.2 A User Updates his/her Profile

(1) The user sends updated profile details to the *User Registration* service via a registration GUI provided by the *Presentation* service
(2) The *User Registration* service updates information relating to the user by calling:
  (a) the *View* service to regenerate the user's personalised views
  (b) the *Event and Change Notification* service to update the user's ECA rules
  (c) the *Update* service to update the user's personal profile
(3) Profile Update is confirmed to the user.

2.2.3 A User Creates a new Group of Users

(1) The user provides details for the creation of a group profile to the *User Registration* service via a group creation GUI provided by the *Presentation* service

(2) A group profile is created from the details supplied — the profile includes a list of users belonging to the group and the fact that the creator is the Owner of the group and the group profile is then passed to the *Event and Change Notification* service
(3) The profile is sent to the *Update* service for inclusion in the local RDF repository
(4) The *Event and Change Notification* service then registers the standard ECA rules for the group profile and then evaluates existing ECA rules that may be triggered by the addition of a new group profile
(5) Creation of the group is confirmed to the user.

The sequence diagram for this use case will be very similar to that shown in Figure 5 for user registration, the only differences being that the user must already be logged in to create a group and that it is a group profile rather than a user profile that is generated and stored.

2.2.4 A User Joins or Leaves an existing Group of Users

(1) The user sends a join/leave group request to the *User Registration* service via a "My Groups" GUI provided by the *Presentation* service
(2) The *User Registration* service updates the group profile by adding/removing the user from the members list using the *Update* service (via the *Event and Change Notification* service, so that any actions triggered by the change to the group profile can be carried out by the ECA engine — see [29])
(3) Addition to/removal from the group is confirmed to the user.

This series of actions will essentially be the same for all other use cases involving the update of a group profile, such as the owner transferring ownership of the group to another group member.

2.2.5 A User or Group Registers an RDF Schema Describing a Taxonomy

The enactment of this and several of the following use cases involving registration of items with a SeLeNe will be almost identical for registrations performed by individual users and for those performed by members of a group on behalf of the group. The main difference in each case is simply that for groups any member of the group can play the role of the actor in the use case but the group, rather than the individual user, is recorded as the Owner of the item registered.

The registration process of a new RDF schema describing a taxonomy by a user or by a group proceeds as follows:

(1) The user submits the new RDF schema via the *Presentation* Service
(2) The *Presentation* Service passes the information, in the appropriate format, to the *Syndication* service
(3) The *Syndication* service performs the registration, using the *Update* service to commit it into the appropriate repository
(4) Confirmation is cascaded back to the user.

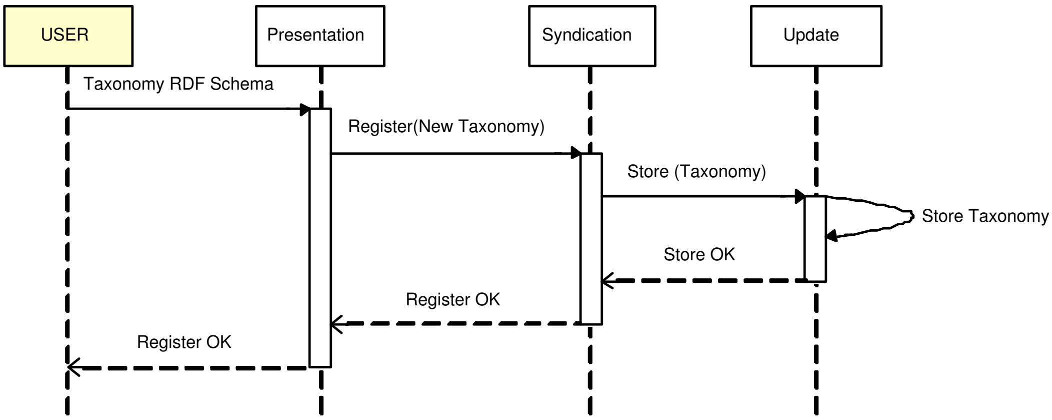

Figure 6: Registration of an RDF Schema Describing a Taxonomy

This is shown in Figure 6.

2.2.6 A User or Group Registers an *Articulation* between a Taxonomy they own and Another SeLeNe Taxonomy

The enactment of this use case proceeds in exactly the same way as that for registration of a taxonomy (Section 2.2.5 and Figure 6). Both taxonomies and articulations are registered via the *Syndication* service.

2.2.7 A User or Group Defines a View over the SeLeNe LO Descriptions

(1) The user submits the description of the view he/she wants to create through the *Presentation* service
(2) The *Presentation* service translates the high-level user request into the appropriate format for the *View* service, for example an RVL Query
(3) The *View* service, using facilities provided by the *Query* service, processes the request
(4) If the view is to be materialised then the *View* service asynchronously contacts the *Event and Change Notification* service to create view-maintenance ECA rules
(5) The *View* service contacts the *Update* service to commit the new view to the view repository
(6) Confirmation of view generation is returned to the user.

The sequence of actions is illustrated in Figure 7. In this and all subsequent enactments that use the *Query* service, **calls from the *Query* service to the *Access* service are not shown**, as *Query* will always need *Access* to the RDF descriptions.

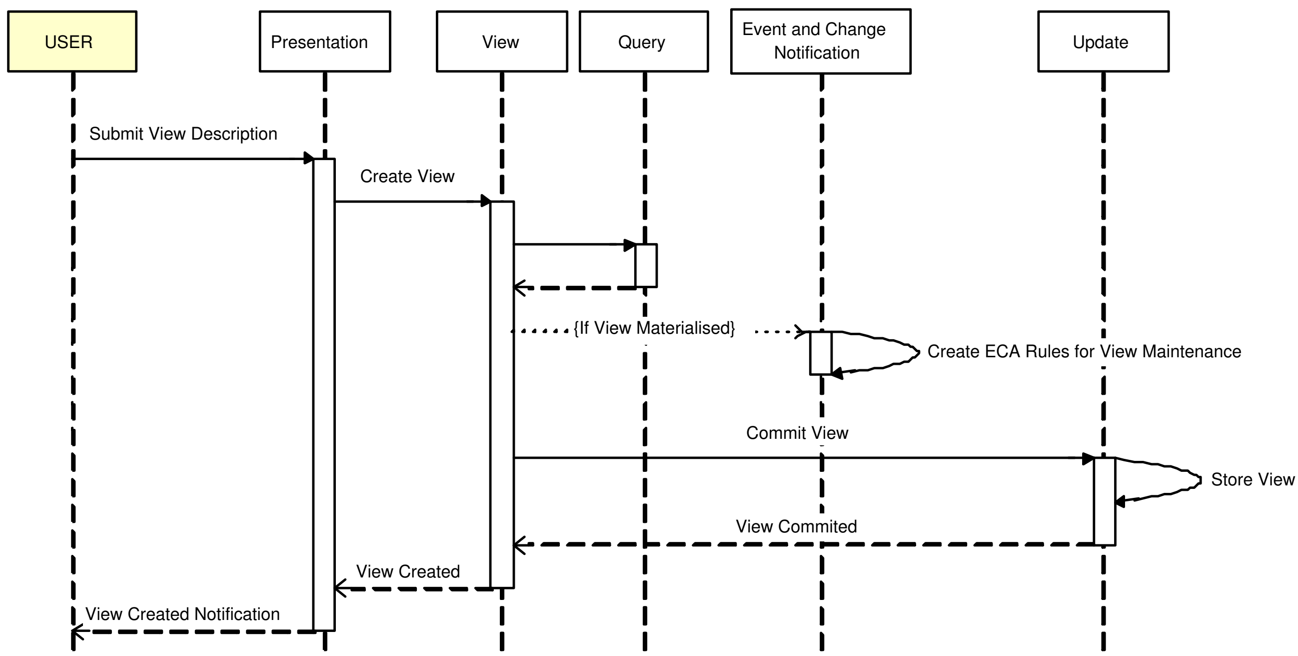

Figure 7: Generation of a new View According to a User's Input

2.2.8 A User or Group Generates a View over the SeLeNe Descriptions based on their Profile

As well as explicitly defining views of the LO information space, views can be generated based on a user or group profile. The enactment of this use case will follow almost the same sequence as that for definition of a view (Section 2.2.7 and Figure 7). The first two stages will instead be:

(1) The user requests a personalised view through the *Presentation* service
(2) The *Presentation* service passes the request to the *Trails and Adaptation* service that retrieves the user or group profile and generates an RVL Query for the *View* service.

Thereafter the same sequence is followed as for view definition.

2.2.9 A User or Group Requests Automatic Notification Services

Users or groups can choose to be notified of changes in SeLeNe's RDF descriptions. The types of change that users may wish to be notified of could be:

- users joining or leaving the SeLeNe
- groups being created
- LO descriptions being added to the repository
- LO descriptions being updated.

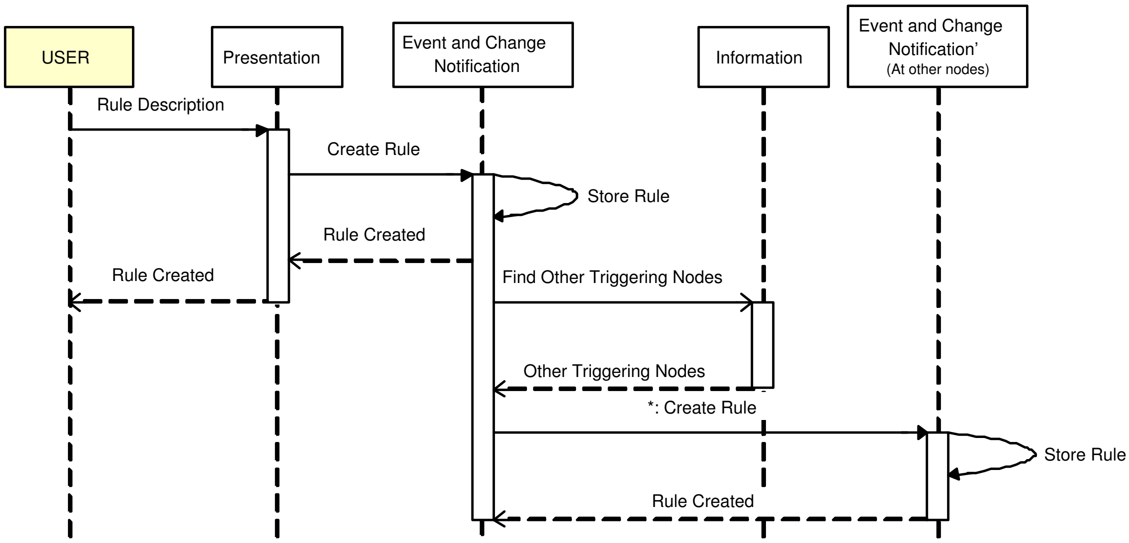

Figure 8: Automatic Notification Request

In all cases, the *Presentation* service generates an ECA rule based on the user's input, and contacts the *Event and Change Notification* service to store the rule. When it fires, this rule will notify the user (or group) by automatically updating their profile with a notification message that can be accessed the next time the user or a group member logs in to the system (see [22] for details of how these messages are stored in the profile). The steps to enact this activity are as follows:

(1) The user submits a high-level description of the automatic updates required through a user interface provided by the *Presentation* service
(2) The *Presentation* service reformulates the user request into an ECA rule (expressed in an appropriate ECA Rule language) [28, 29], and submits it to the *Event and Change Notification* service for processing and storage in the local ECA rule base
(3) The *Event and Change Notification* service uses the *Information* service to find out which other nodes in the SeLeNe hold information that is relevant to the new ECA rule
(4) For each such node the rule is sent for registration with the *Event and Change Notification* service that monitors that node, and each part of the rule is annotated as appropriate (see [29]).

Figure 8 illustrates the steps followed.

2.2.10 A User Registers an Atomic LO with the SeLeNe

NOTE: LO Registration can only take place if certain constraints are satisfied [30].
For atomic LOs, the description supplied in (1) below must be nonempty.

(1) The user, interacting with *Presentation* service, submits details of the description of the atomic LO to be registered via a LO Registration interface
(2) The *Presentation* service collects the information and passes it, in RDF form, to the

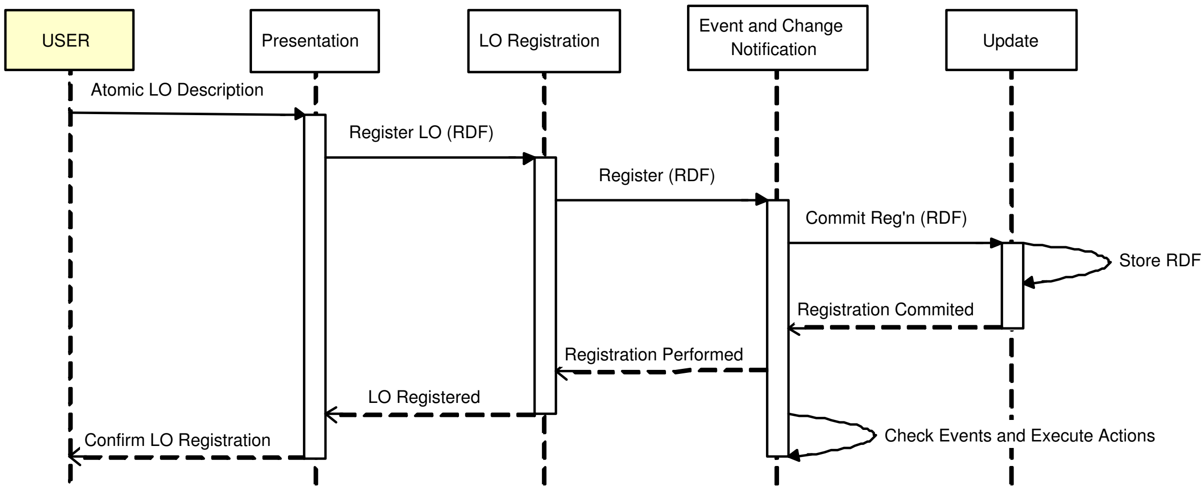

Figure 9: Registration of an Atomic LO

LO Registration service
(3) The LO Registration service then requests storage of the description via the *Event and Change Notification* service
(4) The *Event and Change Notification* service passes the description to the *Update* service for storage
(5) The *Event and Change Notification* service then checks if the new LO description triggers any ECA rules (if so it executes them)
(6) Confirmation of registration is cascaded back to the user.

This is illustrated in Figure 9.

2.2.11 A User Composes and Registers a Composite LO

NOTE: LO Registration can only take place if certain constraints are satisfied [30].
For composite LOs, every atomic component of the composite LO must already be registered.

(1) The user submits the description of the composite LO to be registered, including the 'author description' [30] and identifiers of the component LOs that it is composed of, via the *Presentation* service
(2) The *Presentation* service collects the information and passes the component identifiers to the *Event and Change Notification* service to generate the 'implied description' [30] of the new composite LO from the descriptions of its components[3]
(3) The implied description is returned and the *Presentation* service merges this with the author description to produce the 'registration description' [30] of the new composite LO

[3]We assume that a generic set of ECA rules that take as input a set of LO descriptions and give as output the implied description of the composite LO are available to each Event and Change Notification service. For details of the algorithm used to derive implied descriptions see SeLeNe Deliverable 4.1 [30].

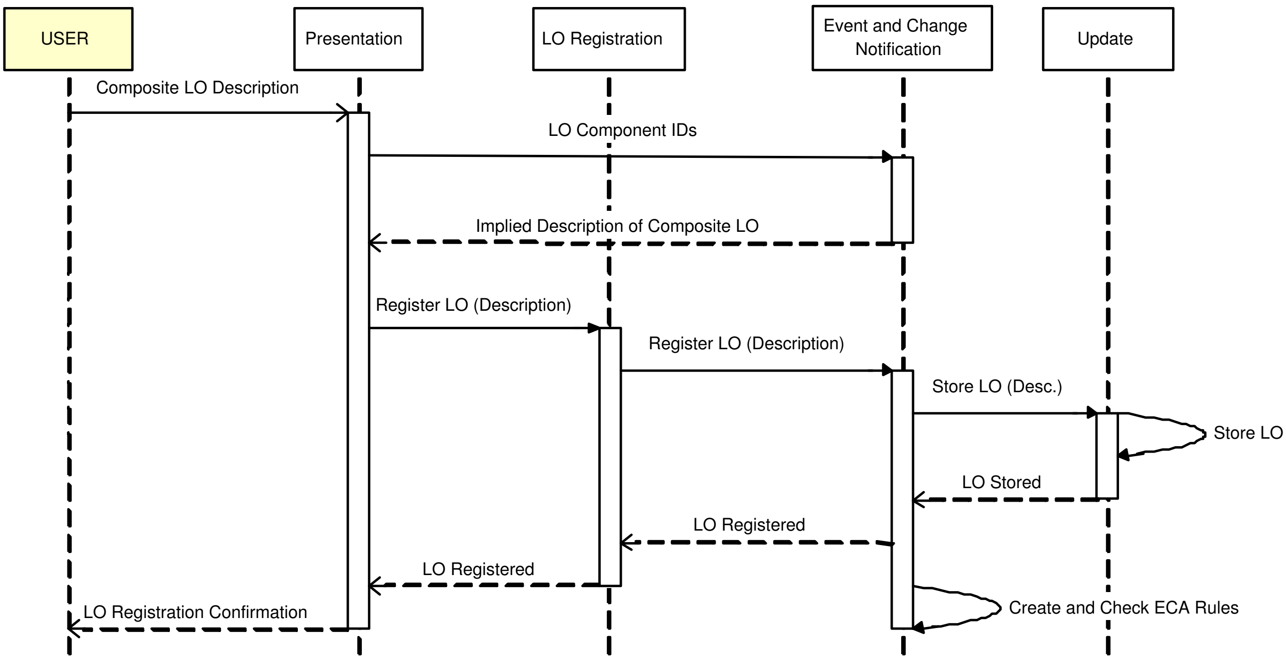

Figure 10: Registration of a Composite LO

(4) This is passed to the *LO Registration* service which then stores the description via the *Event and Change Notification* service
(5) The *Event and Change Notification* service:
  (a) passes the description to the *Update* service for storage
  (b) creates the appropriate maintenance ECA rules for the new LO
  (c) checks if the new composite LO description causes any ECA rules to fire (if so it executes them)
(6) Confirmation of registration is cascaded back to the user.

Figure 10 illustrates the process.

2.2.12 A User or Group Authors and Registers a Trail

(1) The user notifies the *Presentation* service that they wish to author a trail
(2) The *Presentation* service provides a trail-authoring GUI that provides an interface to the *Query* service as well as a trail-authoring interface
(3) The user finds LOs for the new trail using the *Query* interface and creates a trail of LOs, details of which are submitted to the *Trails and Adaptation* service[4]
(4) The *Event and Change Notification* service is called to generate the implied description of the trail as a composite LO (as described in [30])
(5) The registration RDF description for the trail is generated by the *Trails and Adaptation*

[4]If the trail is to be *materialised* then the trail author can save it locally and make it available at a URL that is passed to the *Trails and Adaptation* service along with the rest of the trail details.

service. The description includes:

(a) The automatically generated description of the trail as a composite LO

(b) Additional user annotation

(c) The fact that the user or group is the Owner of this trail

(6) The registration description is stored in the local repository using the *Update* service

(7) Registration of the new trail is confirmed to the user.

This is shown in Figure 11.

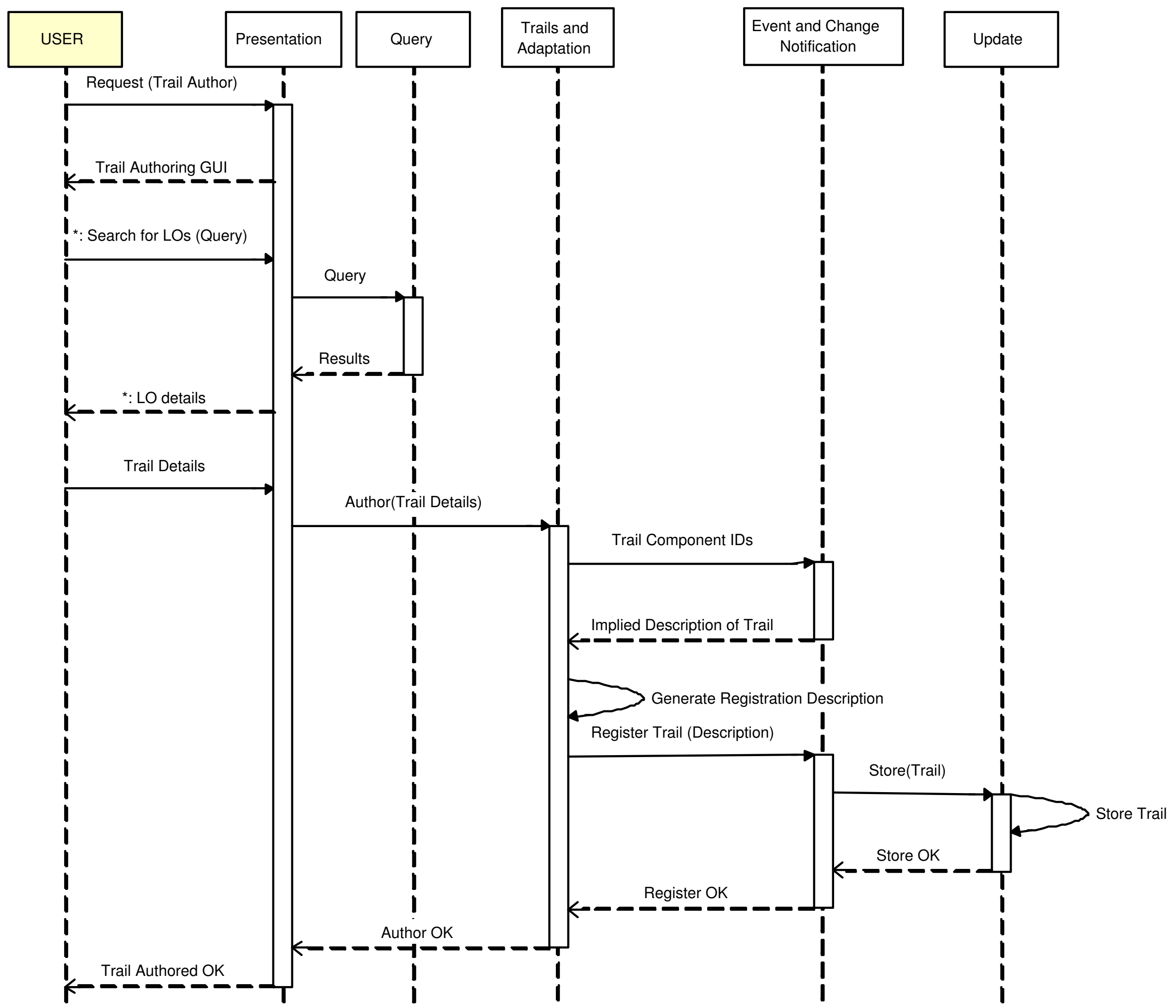

Figure 11: Trail Authoring

2.2.13 A User Requests a Replacement of the LOM and/or the Taxonomical Description of a LO

(1) The user submits updated details of the LO to the *Presentation* service via an 'Update Repository' GUI

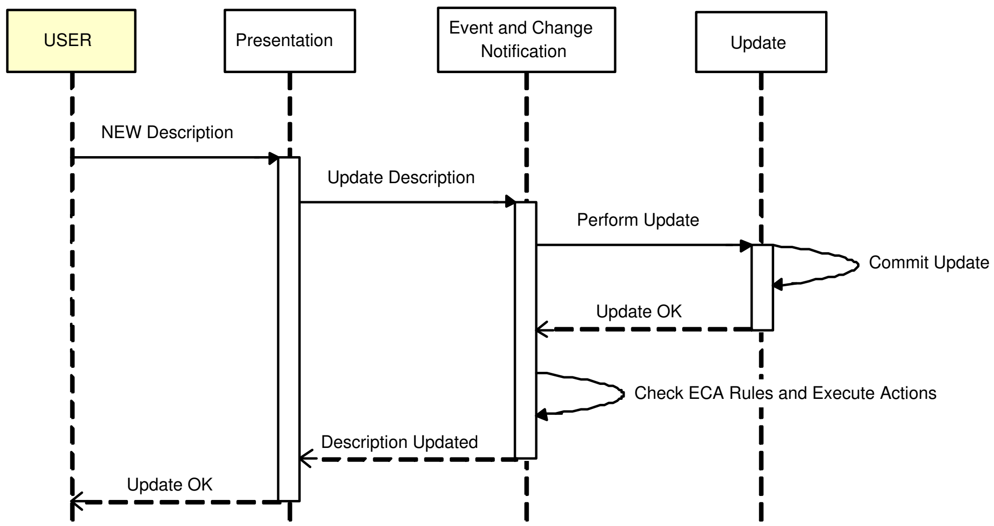

Figure 12: Update of the Description of a LO

(2) The *Presentation* service passes the details to the *Event and Change Notification* service
(3) The *Event and Change Notification* service contacts the *Update* service to perform the update
(4) It then checks for ECA rules that may fire as a result of this update, and executes such rules — these will include actions such as notifying users of the LO whose description has been updated and the re-computation of 'registration descriptions' of Composite LOs that have the affected LO as a component

Figure 12 illustrates the whole process.

2.2.14 A User Withdraws a (Composite or Atomic) LO which he/she Owns

(1) The user tells the SeLeNe that they are withdrawing the LO via a "Remove LO" GUI provided by the *Presentation* service
(2) The *Presentation* service passes this request on to the *Event and Change Notification* service for processing
(3) The *Event and Change Notification* service uses the *Query* service to determine which, if any, users and LOs have descriptions that link via some path in the RDF descriptions to this LO's URL
(4) All ECA rules whose definition specifically refers to this LO's URL are removed
(5) The LO's description is removed from the repository
(6) Notification is sent to the users identified in (3), telling then that the LO has been withdrawn
(7) The notified users then have the option of using a suite of utilities provided via the *Presentation* service to 'repair' their views, composite LOs and trails that have been affected by the withdrawal, using the available RDFS information to remove paths leading to the withdrawn URL.

2.2.15 A User Browses the LO Information Space

There are two closely related use cases here:

1. browsing the SeLeNe RDFS descriptions that the user has access to directly
2. browsing the RDFS descriptions generated by one of their personal view definitions

Enactment of the first case is illustrated in the sequence diagram in Figure 13. All browsing of RDF schemas and LO descriptions is mediated by the *Presentation* service, and proceeds as follows:

(1) The user requests a 'Browse' GUI from the *Presentation* service
(2) The user iteratively browses through schema attributes and each time they browse a new attribute of the schema a query is sent to the *Query* service to retrieve sub-elements of and values for the current schema attribute
(3) The user iteratively browses LO descriptions as retrieved by the *Query* service and presented to the user by the *Presentation* service.

NOTE: steps 2 and 3 can take place interchangeably, allowing the user to return to browsing the schema after browsing LO descriptions.

In the second case, where a user-defined view is browsed, there are two possibilities for the enactment, both of which have similar sequences of service calls as that of browsing the RDF schema directly.

The first possibility is to initially materialise the view. In this case the enactment diagram will be the same as that in Figure 13 but the *Query* service will query the View repository rather than the RDF description repository.

The second possibility is not to materialise the view, but instead to generate queries on the underlying descriptions that will return results that reflect the defined view. In this case the enactment sequence will be the same as that in Figure 13 except that in calls to the *Query* service the view will have to be passed in along with the query itself, so that a query reflecting the view can be generated for evaluation by the *Query* service.

2.2.16 A User Searches for LOs

There are two modes of searching in SeLeNe:

1. keyword-based query;
2. term-annotated keyword query.

The possibilities for the implementation of general keyword search and annotated keyword search are detailed in [22], but whatever search solution is used this will be deployed within the *Query* service. The enactment call sequences for each of these modes of querying will be the same, although there will be differences in the detail of the operations performed by each service.

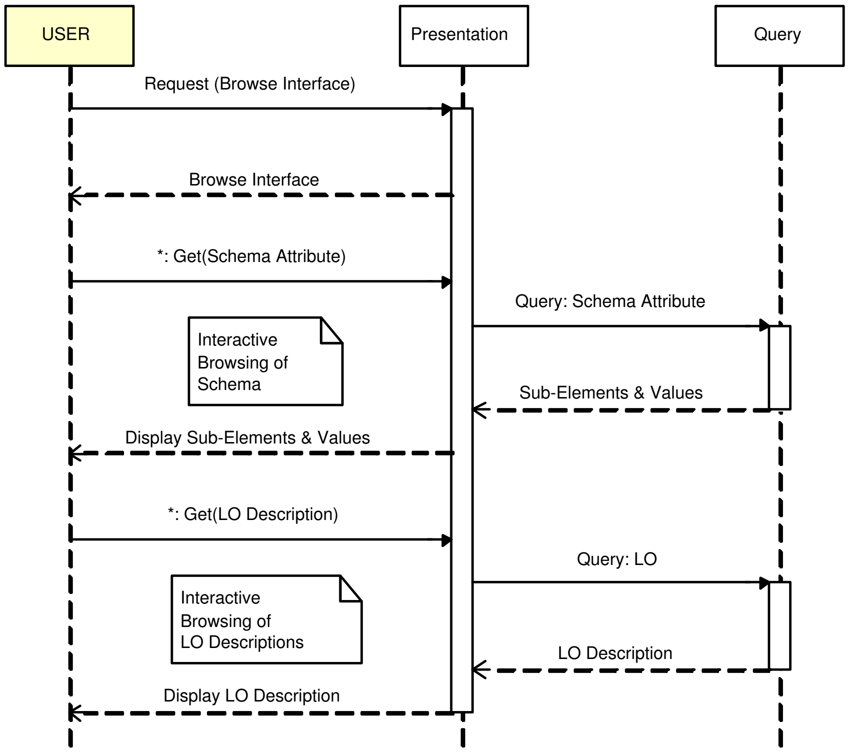

Figure 13: Browsing LOs

For example, when presented with a keyword-based query the *Query* service may perform an Information Retrieval type search across a distributed index but when presented with RQL derived from an annotated-keyword query it will execute the RQL. In both cases the sequence of service calls is identical although the actual operations performed differ.

Similar to the case of browsing, querying in SeLeNe can be done against either the RDF descriptions directly, or against the RDFS schemas generated by a personal view.

The enactment of search of the RDF descriptions directly is shown in Figure 14 and, once the user has accessed the Search interface from the *Presentation* service, it consists of the following steps:

(1) The user submits a query via the 'Search' GUI and the query is passed to the *Trails and Adaptation* service for reformulation according to the user's profile
(2) The *Trails and Adaptation* service retrieves the user's profile, reformulates and extends the query according to information in it (see [22] for details) and passes the reformulated query to the *Query* service
(3) The *Query* service evaluates the query and returns the results to the *Trails and Adaptation* service
(4) The results are filtered and ranked according to the user profile and then returned to the user via the *Presentation* service.

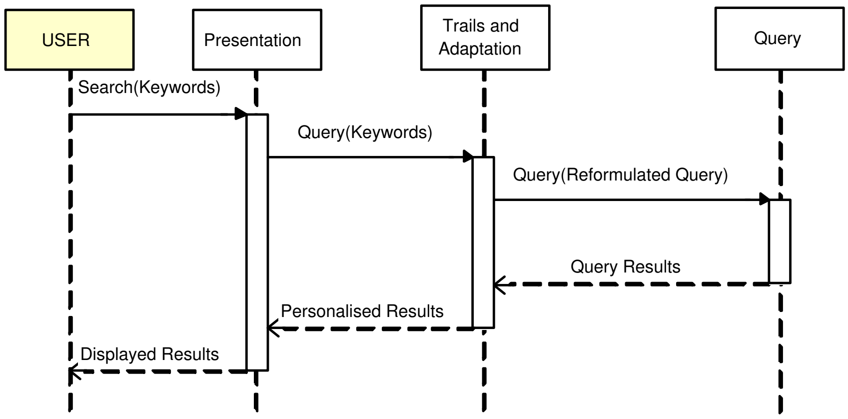

Figure 14: Searching the RDF Repository

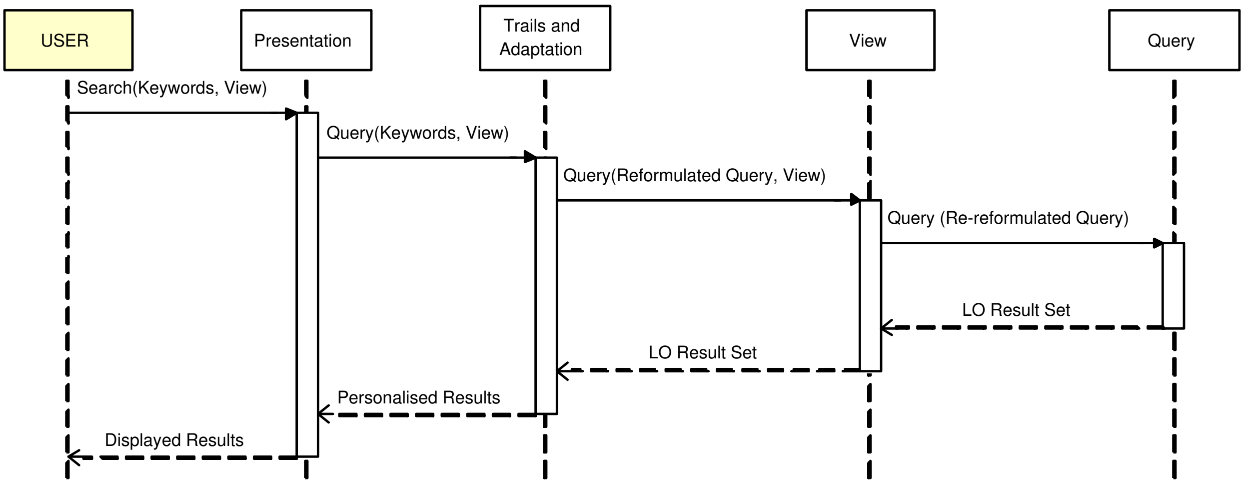

Figure 15: Searching a User-Defined View of the RDF Repository

In cases where search is over a user-defined view rather than the RDF repository directly the enactment will be the same except that queries will additionally be reformulated by the *View* service, as shown in Figure 15.

2.2.17 A Group's Owner Shuts Down the Group

(1) The user requests the deletion of the group from the *User Registration* service, via the *Presentation* service
(2) The *User Registration* service arranges for the removal from the system of information relating to the group by calling:
  (a) the *Event and Change Notification* service to remove the group's ECA rules
    — the *Event and Change Notification* service will first execute a rule that notifies all group members that the group has been shut down
  (b) the *View* service to remove the group's personalised views

(c) the *Update* service to remove the group's profile
(3) Shut down of the group is confirmed to the user.

2.2.18 A User Leaves the SeLeNe

(1) The user requests de-registration from the *User Registration* service, via the *Presentation* service
(2) The *User Registration* service arranges for the removal from the system of information relating to the user by calling:
(a) the *View* service to remove the user's personalised views
(b) the *Event and Change Notification* service to remove the user's ECA rules
(c) the *Update* service to remove the user's personal profile
(3) De-registration is confirmed to the user.

2.3 Use Cases for a SeLeNe System

The use cases for a SeLeNe system are as follows:

- the system generates a recommended trail for a user or a group;
- the system automatically updates a recommended trail, depending on changes in LO or user descriptions, or access history;
- the system automatically maintains materialised views in synch with base RDF/RDFS descriptions.

2.3.1 The System Generates a Recommended Trail for a User or a Group

Trails that are generated by the system can be:

- derived from LO and user/group descriptions
- emergent from histories of LO accesses.

The enaction sequences are the same for both cases, the only difference being the data required by the *Trails and Adaptation* service to produce the trail. In the first case a set of LOs and a profile (user or group) is needed. In the second case only a profile is required as the history of LO access is recorded therein.

The sequence of events proceeds as follows:

(1) The *Trails and Adaptation* service receives an instruction to generate a trail. This could come from several sources — for example, it could be an explicit request for a trail from the user or a request from the *Presentation* service (to organise some query results into a trail, say)
(2) The *Trails and Adaptation* service collects the data required in order to generate the

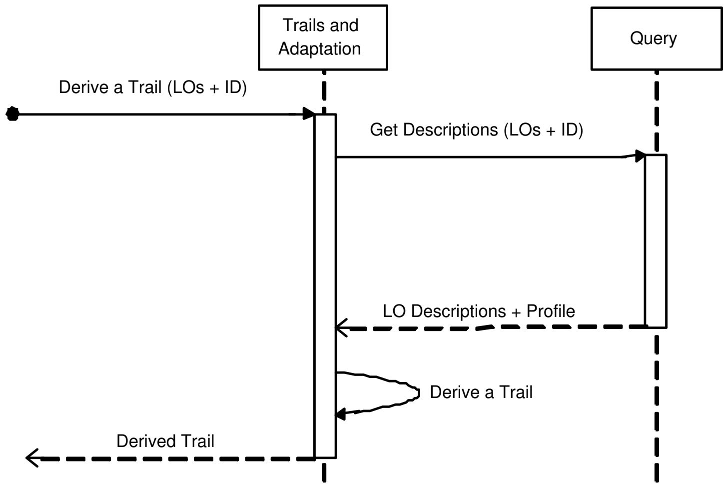

Figure 16: Generation of a Derived Trail

trail, by sending the relevant queries to the *Query* service
(3) It then generates the trail and returns it to the original caller.

This is shown in Figure 16 for the case of a derived trail. The only difference in the diagram for an emergent trail is that no LO descriptions are retrieved, only the profile.

2.3.2 The System Automatically Updates a Recommended Trail, Depending on Changes in LO or User Descriptions, or Access History

Supposing a user action modifies some LO or user description, then the recommended trail update is performed by the following steps:

(1) The *Event and Change Notification* service receives the update
(2) It sends the update to the *Update* service to be executed
(3) It then determines if any rules have fired as a result of this update and if so it executes them

The steps above are illustrated in Figure 17.

2.3.3 The System Automatically Maintains Materialised Views in Synch with base RDF/RDFS descriptions

The case of the view maintenance is very similar to that of trail update above. The *Event and Change Notification* service checks to find out which rules related to view maintenance are triggered by the update/change to RDF/RDFS descriptions. The rules that have fired are scheduled for execution with respect to the materialised views to change/maintain the state of the affected views.

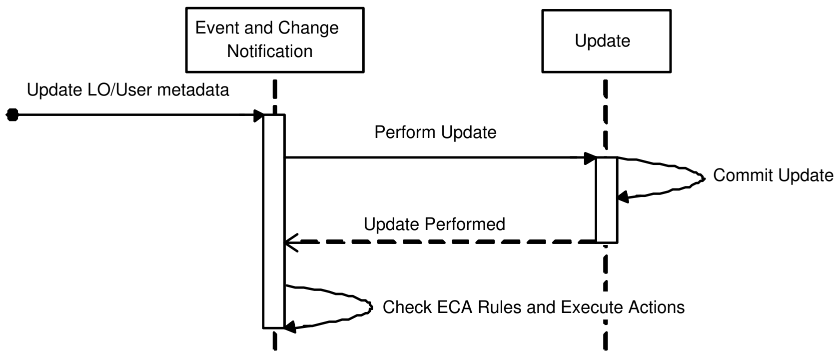

Figure 17: Automatic Update of a Recommended Trail

3 Deployment Scenarios for the SeLeNe Service Architecture

George Samaras, Kyriakos Karenos, Eleni Christodoulou
University of Cyprus

3.1 Selecting the Service Placement Architectural Model

Although building environments for assisting and enhancing the learning process is not a new area of research, technological advancements and especially widespread access to the internet have boosted the interest in creating learning systems that exploit the benefits provided by these new technologies and standards. E-learning is being widely explored [2, 6, 7, 8, 9, 11] and in each case special or specific issues are addressed, as e-learning covers a wide range of different requirements and is incorporated in different applications. These include Learning Management Systems (LMS) (on-line or otherwise), brokering systems for LO discovery and delivery, personalisation agents, resource sharing for scientific purposes (i.e. e-science) and so on. In order to satisfy particular needs for an e-learning system, a development team will define the system components that are necessary to support the system's requested and required functionality. These components then need to be deployed in a specific environment. The internet has allowed today's systems' degree of distribution to grow rapidly, and thus e-learning environments have evolved from centralised, single-point-of-entry concepts to explore ways of realising open and global configurations. In this sense, the placement of components — or in the case of Grid-based systems the placement of services — is not a trivial task. In the SeLeNe project, while defining services for supporting learning environments based on LO metadata, we observed that, although working on this specific aspect, different related projects have assumed or proposed different service placement configurations leading to different architectural models. Additionally, we looked at a learning environment through Grid lenses in order to investigate whether today's Grid technology can be utilised to construct and maintain such a system. Therefore, instead of trying to build a completely new system, we have investigated how to exploit outcomes of previous work, especially efforts related to the Semantic Grid [31]. We have not gone into low-level deployment details, such as describing how to implement SeLeNe's services over Globus. However, as can be seen from the following discussion, *additional SeLeNe services for metadata management are needed, extending and complementing the currently available Grid services, since not all of SeLeNe's services can be constructed using pre-existing components* [5]. More particularly, we propose the incorporation of semantic descriptions of resources through standards such as RDF. Towards this goal, e-learning is an excellent test-bed area.

A similar approach has been adopted by the myGrid project [20], currently in progress, which attempts to utilise the existing Grid infrastructure to create a middleware layer — a toolkit — that will allow scientists to run and monitor experiment workflows in a distributed environment over a resource-sharing Grid. The myGrid project makes extensive use of the Semantic Web technologies and anticipates a migration to OGSI services. It has also shifted from using Web Services metadata storage techniques (UDDI) to RDF, showing the increasing importance of RDF and the flexibility it can provide to such systems. Metadata play an important role in the project and their usage is extensive. In the SeLeNe project we have concentrated on providing services for metadata, targeting the field of e-learning rather than e-science. Also, we do not consider the service ontology aspect (work on this aspect can also be found in myGrid project) but how LO metadata services can be used and combined.

It is worth mentioning that although the target resources are somewhat different in myGrid and SeLeNe (as one project aims to support e-learning and the other e-science), there is still an overlap on the issue of metadata usage as in both cases there exists a great need for querying, personalising and syndicating these descriptions. The investigation of metadata usage to support Grid applications is one area where SeLeNe contributes, and it is on this aspect that the main idea for this architectural model focusses. In the following paragraphs we present the proposed infrastructure for e-science, termed the Semantic Grid [31]. This high-level architectural approach is used to bridge Grid and Semantic Web technologies, and we believe it can act as a model for the exploration of architectural choices in SeLeNe.

Three components comprise the Semantic Grid service-oriented architecture: Service owners, Service consumers and Market owners. A set of service consumers request and use services provided by service owners under specific contracts that define the conditions under which the provision is allowed. As these links are established, a marketplace is formed, i.e. an environment under a specific set of rules for its participants. This marketplace is run by the market owner, which defines the rules. Marketplaces can either be private or open to further participation. In the case of the SeLeNe project, the marketplace is actually some Self e-Learning Network that is established due to the presence of a set of services that are defined in Deliverable 3, which constitute a functional set for this system. The

[5]The term "component" here refers to a unit or group comprising units that provide some functionality to the system either these being services, user agents, processing components, etc.

data exchange "rules" [31] are defined by the service API's. However, other such "rules" may mandate the network (e.g. security policies) that can also take the form of services but that are not addressed in this project[6]. Authorities [32] provide services that at a minimum ensure the ability of a participant (provider or consumer) to enter a SeLeNe. In this sense they act as the entities that establish a SeLeNe.

We note that just as multiple marketplaces can be created, similarly multiple SeLeNes can be formed. By this we mean that services provided by a specific participant in an established SeLeNe can also form part of another SeLeNe, possibly established by another Authority.

We also reiterate that SeLeNe services are metadata-oriented. In the Semantic Grid description the notion of services provided for consumption is broader and more generic. However, we attempt to establish the relationship between SeLeNe and the Semantic Grid to illustrate that the inclusion of our service set under the scope of the Semantic Grid is possible — for example, having RDF metadata services introduced as a subset of the global e-science service set (i.e. to support the workflow creation process).

Figure 18 illustrates the correspondence between the Sematic Grid and SeLeNe. There are two important characteristics that should be highlighted with respect to this figure:

- SeLeNe can be seen as an application of the Semantic Grid. A SeLeNe corresponds to a "marketplace" consisting of service consumers and producers. In our case, the SeLeNe initiator is termed an "Authority" and corresponds to the marketplace owner. The marketplace/SeLeNe is built from a number of available services. In e-science any possible service instance can be part of the marketplace, whereas we specify exactly a minimal set of services that need to be present for the formation of a SeLeNe (see Deliverable 3 [32]).

- One of the characteristics of the Semantic Grid is the lack of restrictions as to the roles that a site may acquire (as mentioned below). In Figure 18 the separation between consumers, providers and Authorities is mainly conceptual — a SeLeNe site can interchangeably play the roles of consumer or provider. The distinction is made in order to clarify the communication among the services available at each site, and to better demonstrate the relevance of SeLeNe with respect to the Semantic Grid functionalities described in [31].

In Deliverable 3, we have also established the connection with respect to the Semantic Grid's three levels of abstraction (Data–Information–Knowledge). The justification of selecting the Semantic Grid approach as the abstract model for the SeLeNe service placement architecture is summarised by the following points:

- The Semantic Grid infrastructure agrees with our service-oriented approach and provides a good justification of why we can adopt such an approach. As detailed in [31], the approach provides a separation among service providers, consumers and mediators as well as a model for establishing how they are related to one another. The

[6] See Deliverable 3 for additional research issues that have been identified.

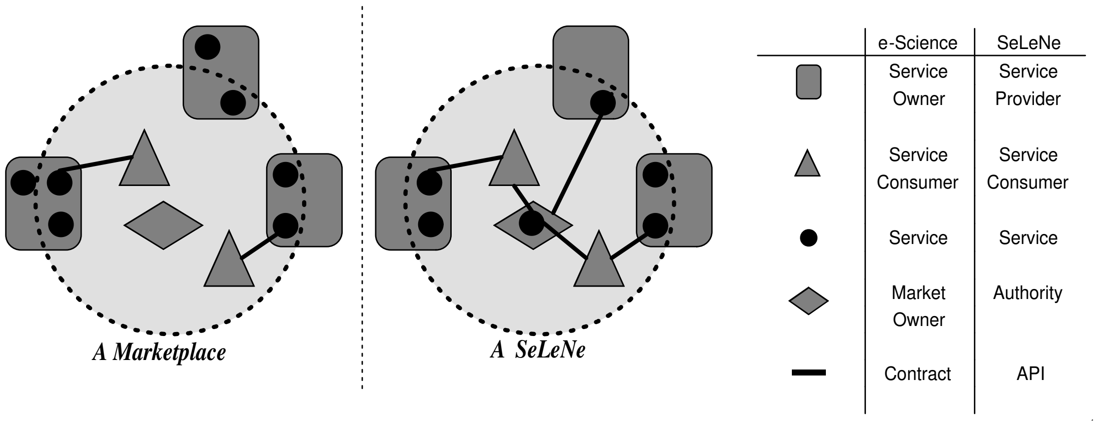

Figure 18: Relationship of Semantic Grid and SeLeNe

functionality of a system is realised by looking-up and combining a number of services. When this functionality is a joint effort (such as in an e-learning environment), these services are provided by the entities participating to the learning process. An additional advantage of this model is that marketplace owners (corresponding to Authorities) can be either service providers, service consumers or a third party. For example, Figure 18 does not depict topological architectures. This notion is also reflected in the "Architectural Alternatives" section.

- The Semantic Grid infrastructure is generic enough to include our proposed service set. Service owners and service consumers are not restricted to these roles and may interact accordingly. Although interaction is assumed to be carried out by agents, this is not a limitation and neither is it imposed. Therefore, if the set of interacting services that fulfil the system requirements is known, such as in SeLeNe, this model is still applicable. The "Service Lifecycle" process (creation–procurement–enactment) makes no assumptions as to the steadiness of the system. Therefore, different architectural approaches are possible: from static (e.g. centralised) to dynamic (autonomic P2P).
- The Semantic Grid infrastructure was proposed and included as part of the GGF and thus we expect that it will receive higher priority over other proposals while Semantic Grid research is moving towards the definition of standards.

3.2 Emerging Issues

Throughout our work towards deciding and defining an architectural model for SeLeNe, we studied the current Grid Infrastructure and its completeness with respect to supporting a metadata-based, e-learning environment. In doing so, we have identified some areas in need of improvement or further exploration to complement the functionality provided by current Grid technologies. These areas are discussed in the following sections.

3.2.1 Semantic Descriptions for the Grid

Metadata has been widely used in a considerable number of large Grid and e-science projects (see [20, 4, 1, 3, 5, 10], for example). Depending on the application area, metadata is used in different ways. EGSO, for example, attempts to produce a catalogue for solar observation data under a common metadata standard for solar physics. DataGrid also utilises metadata cataloguing schemes to locate and manage large data sets. These projects, and especially DataGrid, are characterised as "traditional Grids" and no use of semantics is explicitly made. A method for creating commonly understandable metadata to describe resources and to define uniform ontologies for each application domain would have been greatly beneficial, as would predefined services that provide mechanisms for bringing metadata to use by each application. To date none of the works cited have proposed a standard for metadata usage. It is, therefore, important that *standardised semantic models* become available. In SeLeNe we have worked towards that direction by utilising RDF.

The benefits of standardising resource descriptors to be used as metadata have a number of advantages. In the Access Grid [1] project for example, which has demanding multimedia requirements, metadata services could provide the means for discovery of resources, combination of multiple resources, mappings of resources to applications, etc. Goble and De Roure, in their work on "Semantic Web and Grid Computing" [19], reiterate the statement that the Grid can be considered an application for the Semantic Web and they provide an enumeration of areas where Semantic Web technology can be applied to the Grid. Below we list a number of such opportunities which are also relevant to the SeLeNe project, i.e. with respect to a data-centric rather than process-centric approach. As discussed in Deliverable 3, though, data-centric approaches do not cancel out process-centric ones. We add to these opportunities, based on our own work in an e-learning application.

- *Metadata-based Middleware*: Metadata play an important role in many Grid projects and thus, Semantic Web technologies can be applied for this purpose. The basic issue here is the diversity among various application domains. However, this is a central goal of the Semantic Web, i.e. finding a common way of describing the resources, providing means for their intercommunication and linking among their functionalities. The development of metadata-based middleware is not application-dependent, and as such it can be utilised by any one of these applications. In SeLeNe we have proposed a set of services, some generic and some specifically to provide the functional requirements of SeLeNe. These are based on Semantic Web technology, and more specifically RDF, to support querying, syndication, automatic notification, personalisation and trail maintenance at the application level.

- *Dynamic Combination of Resources*: Brokering diverse resource descriptions is a capability offered by the Semantic Web even when the same resources are described under different ontologies. Integration is a common vision of both the Grid and the Semantic Web.

- *Collaboration*: Two of the goals that may be required by an e-learning environment can be achieved: (i) browsing through the resources on a remote site (or multiple combined sites i.e. "information islands") with which a collaboration agreement has been established, and (ii) multiple participants combining multiple resources to build new materialised or virtual LOs. In these two scenarios the use of metadata-based services is a primary requirement.

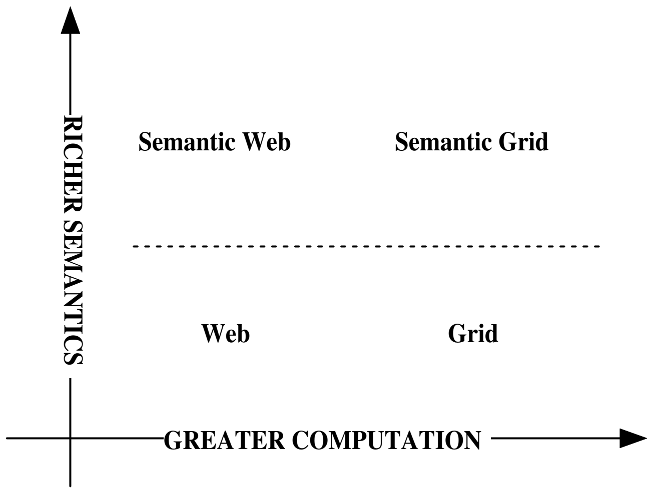

Figure 19: Semantic Web and the Grid

The reason for the lack of semantic metadata usage seems to have been the fact that Grids are primarily process-oriented. This dimension is orthogonal to the richness of semantics provided by Semantic Web technology (see Figure 19, which is taken from [19]). Deliverable 3 [32] also includes a relevant discussion on Grid Services.

In general, there are three broad areas that the Semantic Web can assist in e-learning: resource matching (e.g. via personalisation), service matching (e.g. via definition of application ontologies) and reconciliation of diverse LO descriptors via mediation mechanisms([25, 33, 24], and SeLeNe Deliverables 4.1 [30] and 4.3 [26] on Syndication and Querying).

3.2.2 The Case of the Globus Toolkit

Our discussion of how SeLeNe services should extend or complement the Grid has so far been rather generic. In order to give an idea of how existing Grid resource description and discovery services could possibly be extended through the use of metadata, using SeLeNe's generic services, we consider the Globus Toolkit [12] and its resource description techniques.

The most relevant Globus components for resource discovery and Grid services' information are the Globus *Information services* [17, 12], which are also known as the *Monitoring and Discovery Service (MDS)* [13]. MDS stretches across the Connectivity, Resource and Collective OGSA layers, as was our goal when defining SeLeNe's generic services. The basic MDS hierarchy of components and protocols is as follows (for more details and updates please see [13]):

- *Information Providers (IP)* are the base components that directly interface to a resource;

- *Grid Resource Information Service (GRIS)*, which is contacted by IPs, runs on a specific resource and acts as that resource's content gateway;

- *Grid Index Information Service (GIIS)*[7], which is responsible for allowing GRIS registering resources. GRIS may register to *any* GIIS and any GIIS may register with another GIIS;

- Two base protocols allow for the communication between Grid Information Service sites: *The Grid Information Protocol (GRIP)* is used to access information about entities, and the *Grid Registration Protocol (GRRP)* is used for registration and information availability notification to the directory services.

There is an important difference between MDS and SeLeNe's metadata services. MDS functionality is based on extensions to the LDAP standard and is better suited for computational processing resources such as CPU availability, disk space, etc., as well as for identifying service state using Service Data Elements (SDEs). Therefore, no semantic metadata descriptions can be easily incorporated into MDS. On the other hand, using MDS can benefit possible extensions since it is already part of a complete toolkit that provides essential core functionalities and, more specifically, concrete security infrastructure (i.e. GSI).

Although SeLeNe's service functionality provides a complete information/metadata toolkit (since it provides complete functionality: registration, query, syndication, adaptation), it is still extremely difficult to claim the replacement of, or even direct integration of, semantic resource descriptions with Globus MDS. One alternative would be to implement SeLeNe services as completely independent entities, i.e. as additional Grid Application, OGSI-compliant Services. Another option would be to have SeLeNe services extend the previously presented MDS architecture. One possibility for the implementation of this (illustrated in Figure 20) would be the following:

- SeLeNe sites act as IPs (where the Information is the descriptions available at local repositories). It is assumed that Core SeLeNe services run on these sites, including the Information and Access services that are essential for this functionality.

- GRIS runs on Authority sites. SeLeNe providers (IPs) register resource descriptions to the Authorities (note that Authorities can themselves be providers). Authorities thus act as "gateways" to the rest of the Grid.

- GRISs register with *any* available GIIS. In this way SeLeNe services are made accessible to external users by querying the GIIS.

[7]GIIS has now been extended by the Globus Toolkit Version 3, *Index Service*.

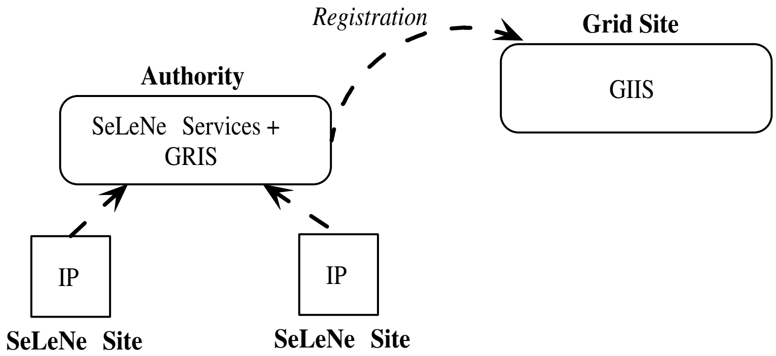

Figure 20: An example of SeLeNe Services over Globus Information Services

Two additional minor modifications are possible: (i) For a more autonomic system, each site may run its own GRIS responsible for the local resources, (ii) Authority sites could also host GIIS and incorporate extended indexing services, which in our case are SeLeNe's metadata services.

3.2.3 P2P Grids

The traditional architectural model for Grids is the client-server one. However, as new Grid protocols are being developed [12], a growing interest in providing support for P2P applications via these protocols is obvious within the GGF. In finding a suitable model for our e-learning environment, which would fit multiple learning scenarios [23], we have come to the conclusion that a single architecture cannot be applicable for every case. We identified a number of SeLeNe characteristics that require P2P support — this was part of Deliverable 3 [32]. Although the feasibility of deploying, for example, different Globus services on a group of sites as well as having those sites also using the client API's (i.e. act as both clients and severs) is not within the scope of our work; such an attempt would probably have extended requirements such as aggregation and control over the provided services. A special research group has been formed at the GGF, the OGSAP2P group, to take on the task of producing a set of protocol requirements and service definitions for P2P Grids. These definitions are to cover security and trust, connectivity, and interactivity. The OGSAP2P group has also defined a taxonomy of P2P architectures [21] based on a number of core operations, some of which we have already looked into in the SeLeNe project. Here is how we see a SeLeNe in relation to these core operations:

- *Identity:* This refers to the name and credentials of a user or peer. In the SeLeNe project we differentiate user login from site login as user adaptation/personalisation is required (i.e. the organisation of nodes is a separate function from the identification of users — these functions are provided by the *Sign-on* and *User Registration* services, respectively). In general, as each peer[8] in such a system is autonomous,

[8]In the context of a P2P/Autonomic architecture, the terms "peer" and "site" may be used inter-

personalisation strategies will be required, thus a single identity for both site and user does not suffice.

- *Discovery:* A broad definition of this is given by the OGSAP2P research group, stating that this operation provides the means of finding out what machines, services, or resources are currently on-line. A clear distinction must be made between service discovery and resource discovery. In SeLeNe we propose two separate services: *Locate* for service discovery and *Query* for resource discovery.

- *Authentication and Authorization:* The identity of a user and a physical site need to be securely identified. In addition, possible privileges and/or restrictions need to be imposed on the resources and especially on invoking a service or a specific functionality[9].

- *Function:* This operation defines the actual functionality of the system (e.g. file sharing). This is again rather broad and quite generic. An e-learning system such as SeLeNe defines Function to be Personalisation and View formulation, Notification of events and Query over the LO metadata space.

In the OGSAP2P group's work different topologies are also proposed depending on the goal of a distributed functionality and interactions among the operations above: client-server, direct, mediated and inverted client-server (see Figure 21). In the SeLeNe project we view the problem of defining a topology rather as a service placement problem, i.e. depending on the target environment different placement configurations can be applied. In SeLeNe we produce a taxonomy of services leading to generic architectures. A SeLeNe is formulated under some topology due to the existence of the service set that achieves the initial system requirements.

3.3 Architectural Alternatives

In this section we provide a family of architectural alternatives for SeLeNe. We will also provide examples to show how the enactment of certain Use Cases will occur at the level of machine-machine interaction for each of the deployment possibilities described. Our goal is not to produce a novel model or provide technical solutions to site organisation but instead, based on the set of proposed SeLeNe services, produce service placement scenarios for which different topologies are produced. The basic idea is to illustrate the formation of one or more Self e-Learning Networks in different learning environments. A "learning environment", from the architectural point of view, is parameterised by the degree of dynamicity in participation and service provision, distribution of resources (i.e. LO descriptions), the actual storage of the description bases and the topology that service providers and consumers create. For example, a collaborative all-peer learning environment cannot assume a centralised storage of descriptions nor can it execute all processes on

changeably.

[9]See Deliverable 3 for open issues related to security in SeLeNe.

descriptors (such as query) at one specific location. On the other hand, for a SeLeNe serving a single institution a centralised repository could be kept of whatever resources the instructors and learners register and make available.

Each site can be a service provider and a service consumer at the same time. Note that this does not imply a pure P2P modelling of the sites. Instead, the storage and processing of metadata will define the degree of independence among sites. One site may need to request a service from another site, but it will generally assume that it can only provide core services. The actual formation of a SeLeNe will depend on the availability of the set of services identified as those that satisfy the user requirements, thus allowing the operation of that SeLeNe.

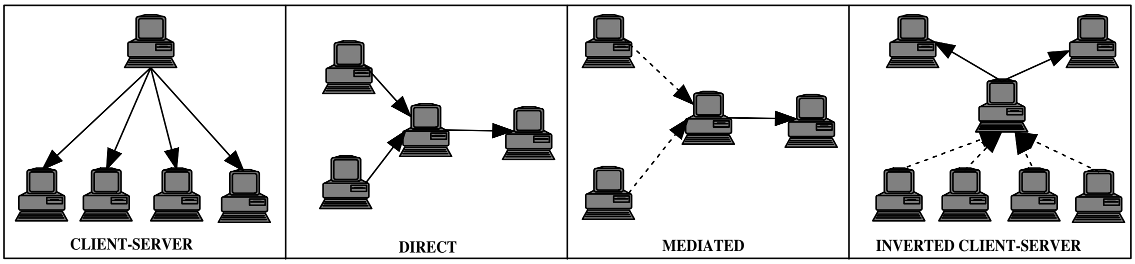

Figure 21: Topology patterns emerging from the system's "Function"

For completeness we will describe briefly the topology possibilities identified in the draft version of the P2PGrid Working Group at GGF on the "Taxonomy of P2P Architectures" [21]. We describe briefly below the "Function" operation, i.e. the flow of services, or "distributed activities" as it is described in the draft, from providers to consumers. In Figure 21, four patterns are shown. Arrows imply direct flow of service from provider to consumer. Dotted lines imply requests that are mediated.

A *Client-Server* topology is found in traditional Grids. A central machine is the only service provider which receives requests from multiple clients. In the *Direct* topology, each machine may provide services, but it does so on its own. The *Mediated* topology allows for services to be syndicated in order to provide the reply to a request. In the *Inverted Client-Server* topology, the central machine farms requests and distributes them to the edge machines, which act as servers.

The relevance of Figure 21 to the alternatives proposed below is that it captures multiple ways in which service interactions may occur. However, as mentioned in the GGF draft, there is not one single way in which services are accessed but in many cases combinations of the patterns may arise. Therefore, a "mediated" access pattern does not necessarily imply a particular network configuration such as Super-Peer or Pure P2P. In this section we are interested not only in service interaction but mainly in service placement as well. We have decided to use the terms *"Centralised"*, *"Mediation-based"* and *"Autonomic"* to describe the characteristics of each SeLeNe learning environment with respect to service placement.

The summary of the proposed SeLeNe services in Section 1 should be used as a reference when reading the subsections below. Examples of the interaction among services is provided for each architectural alternative using two of the use cases from Section 2:

(i) A use case for nodes: A new node joins a SeLeNe (see Section 2.1.1).

(ii) A use case for human users: A user or group generates a view over the SeLeNe descriptions based on their profile (see Section 2.2.7).

We chose these use cases since they are distinctive enough to achieve the purpose of the examples described, i.e. to show differences in terms of service calls for each model that are due to the different topologies.

3.3.1 Centralised

Description The "Centralised" model is similar to the traditional Grid model. It is not exactly the same, as it requires core services to be run by what in the traditional model would be "clients" or "consumers". The centralisation in this case has more to do with the fact that a greater percentage of computation and the totality of the RDF descriptions storage are found at a centralised location. This can be a single machine (which will run all services) or a computational cluster which consists of physically adjacent machines. These machines provide all the required services and will, at times, require access to "client" machines to request access and query services, for example to gain access to the actual metadata. This group of dedicated "servers" is static and interconnected over a fast, local network. Services running within this cluster will probably communicate using some RPC (e.g. RMI) rather than messages (e.g. SOAP). Consider, for example, a similar approach seen in heavily loaded web servers: although one will post a query to `google.com`, or request a dynamic page from `cnn.com`, requests and replies may be processed by any of the machines that handle the web load comprising the web server farm. The content accessed to provide a reply is the same for all requests.

One or more dedicated machines act as Authorities. An Authority acts as an entry point to the SeLeNe. A "client" site will know the Authority site a-priori. This site will run a number of services required for the initiation of participation of a "client" to the SeLeNe, that we will describe below.

Figure 22 illustrates this topology: Machines indicated by 'S1', 'S2' and 'A' comprise the centralised SeLeNe cluster, where 'A' is the Authority site. "Client" sites are indicated by 'P1', 'P2' and 'P3'. A continuous line indicates a static connection among the machines whereas a discontinuous line indicates a connection that can be established for a service request/reply. "Client" sites may communicate with any one of the centralised servers as long as these have been made known to them. However, a new site will need to contact the Authority to discover who can provide the service it wants and will be directed to it by the Authority automatically. If a cluster of service "providers" exists instead of a single machine, this is transparent to "client" sites. Also, "client" sites are not directly aware of other "clients" connected and may communicate with each other via the centralised servers (e.g. by accessing a centrally administered collaboration service).

Metadata is located at the client machines but must be registered to the centralised system. Therefore, the RDF repository is distributed among the cluster machines although it is treated as one common repository by all servers.

Service patterns that can be seen in this model are "Client/Server" and "Direct" (among the servers), as described in the introduction of this section. However, services can be placed in a way such that processing load is shared among the cluster machines which seem to provide a single point of access to the outside observer. We provide concrete suggestions below.

An example of such a topology is that of a single educational institution that builds up a SeLeNe for its members. Participants connect to the central administration machines and register their LO metadata which are also maintained by the central servers. Services such as querying the LO space, creating personalised views and updating can be accessed through the SeLeNe central servers.

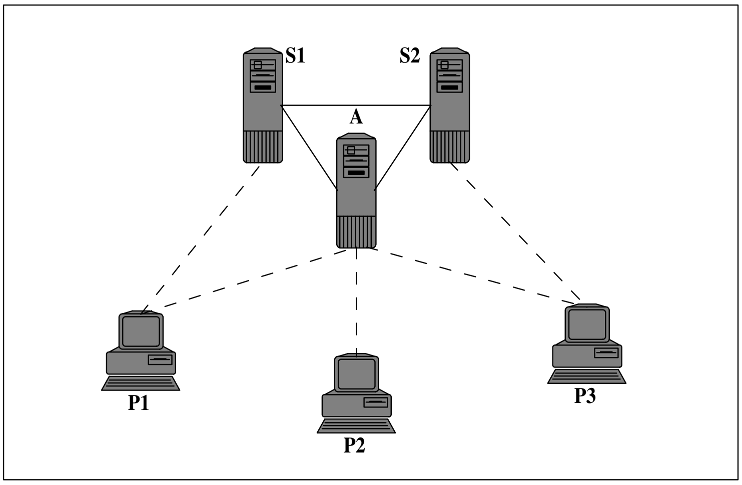

Figure 22: Centralised Model

Services In Table 1 we provide a specific proposal for service placement. We again note that Core services are assumed to be available at every SeLeNe site. These are quite a small set of services and are the only ones explicitly required to be provided by the "client" participants. We also note that a service may be running on multiple different servers and may be called from any of those. The first group of services is found at the Authority site (or put in another way, the site that provides these services will be assigned the role of the Authority). These are important because they allow a site to become part of the SeLeNe and to lookup a service that it requires. The Authority is also recommended to play a central role as a coordinator during the creation of views, by maintaining a global picture of the current LO descriptions space.

A second group of services is recommended to be common to all servers in order to allow better work load distribution. These are the registration services (i.e. a client can register LOs and lookup their description on any central server). Also the *Syndication* service is central to a number of other basic services such as *Query* and *Update*, thus multiple instances of it should be available.

The remaining services are shared among the servers. For example, one server can run the Trails and Adaptation service and keep track of user profiles whereas another can handle collaboration requests.

Examples for the Centralised Model

In the description of service interactions it is assumed that for message exchange and passing of parameters the *Communication* service is used. Thus, for simplicity, we will omit the details of calls to the *Communication* service from the examples, as with the use cases in Section 2. We will, however, include the Communication service for Example (i). Some exchanges that are mentioned in Section 2 (such as confirmations) are implied and will also not be repeated in the following examples.

(i) A new node joins a SeLeNe

Imagine a new client/consumer site wants to join the SeLeNe — for this example we'll assume 'P1'. The services present are shown in Table 1. 'P1' only runs local, core services. 'P1' is aware of only the Authority site 'A', when no prior communication with the SeLeNe has been established. P1 calls for the Locate service at 'A', via the Communication service. It is important to note here that for a *centralised model*, 'P1' could *directly* request for the Sign-on service since this service is available at 'A'. 'A' then replies with "contact information" (e.g. service port) for the Locate service (again, via the Communication service), allowing 'P1' to request the Sign-on service. Location of the services is straightforward since lookup is static.

(ii) A user or group generates a view over the SeLeNe descriptions

This is a rather more complicated case of service interactions. However, since all services and repositories are centralised, these interactions as described in 2.2.7 are, again, straightforward. Authority 'A' will redirect the user to 'S2' which runs a Presentation service in order to allow for the capturing of the user's query. As with example (i), a direct call to the View service, available at 'A', is possible but would require that the view definition is directly included with the service call (e.g. a RVL query). The View service needs to call the Query service. We note here that for the *centralised* case a *single* View service instance (per user), issuing requests to the Query service, is sufficient. Similarly, the Query service, although present at each server (since it is core), does not need to activate the Query service on other local servers. It will complete the querying process by

A	S1	S2	P1, P2, P3
Core	Core	Core	Core
Sign-on Locate View	Collaboration Update ECA	Presentation Update Trails/Adaptation	
Syndication User Registration LO Registration	Syndication User Registration LO Registration	Syndication User Registration LO Registration	

Table 1: Centralised Model Service Placement

accessing the description repositories residing on any server by retrieving data via each server's Access service. On materialisation, the interactions follow the sequence described in 2.2.7, i.e. call to the ECA service and Update service residing on 'S1'. A reply is sent back to the user by 'S2', which runs the Presentation service. If the interaction did not involve 'S2' (e.g. direct call to the View Service) the client must anticipate an answer from 'A'.

3.3.2 Mediation-Based

Description The Mediation-based model has many similarities to the Consumer-Broker-Producer paradigm. Although we will use the terms "Consumer" (or "Client"), "Broker" (or "Authority") and "Producer" (or "Provider") sites, this is done to distinguish among their primary functionality (i.e. services hosted) and degree of dynamicity with respect to the SeLeNe. We provide below the minimal characteristics for each type of participant. For example, although a "Consumer" or "Client" site's primary functionality is to request services from remote sites, this does not mean that it should only run core services. On the contrary, it is expected that despite the fact that most "Consumer" sites may be entering and leaving a SeLeNe in unpredictable patterns, they may provide services available to other sites whenever connected.

The reason for this model to be named Mediation-based is due to the fact that its functionality is primarily facilitated by mediator machines, similar to "Brokers" that we refer to as "Authorities." Authorities are affiliated with a number of "Providers" that become known to them and are characterised as their neighbours. Authorities host a number of specific services required to allow distribution of requests beyond their neighbouring "Providers". On the other hand, "Providers" will host complementary services to realise the formation of a fully functional SeLeNe. Thus a "Consumer" will direct its request to its directly known Authority which will, in turn, direct that "Consumer" to the appropriate "Provider." A "Provider" may be directly known to that Authority or may be discovered via request-forwarding to another Authority. The hierarchy of Authorities is not a basic issue and it mainly depends on the discovery protocol utilised by the Locate Service, thus it does not

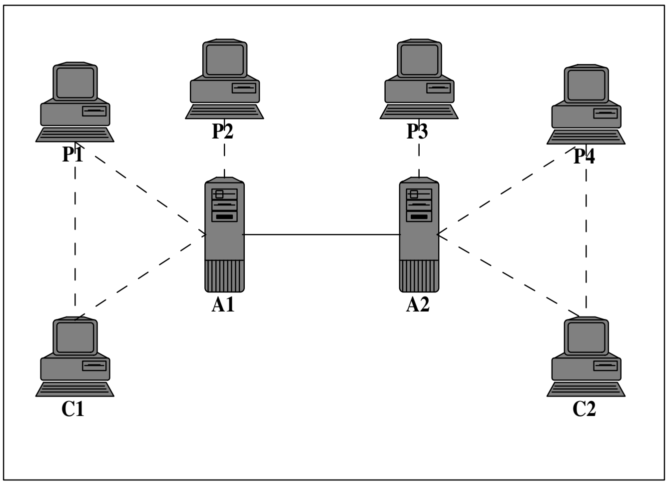

Figure 23: Mediation-based Model

constitute part of our discussion. “Providers”, although less static than Authorities, are typically more reliable than “Consumer” sites.

Figure 23 demonstrates one such setting. ‘A1’ and ‘A2’ represent the Authority sites, ‘P1’ through ‘P4’ represent the “Providers” and ‘C1’ and ‘C2’ represent “Consumer” sites. Connections among Authorities are static and more reliable. A “Consumer” may contact any “Provider”, therefore not all possible links that can be established are shown. Reduced reliability with respect to the Centralised model is compensated for by allowing multiple service instances to be available. In addition to service availability, issues related to data availability (e.g. via caching and replication techniques, as well as construction of communities which share related LOs) are part of our discussion in Deliverable 3. Metadata are still stored at the “Consumers” but again they need to be registered with an Authority. Authorities, therefore, in addition to entry points also carry out two basic tasks: maintaining LO descriptions and User Profiles via the Trails and Adaptation service. Additional functionality is provided by the “Provider” sites in the rest of the SeLeNe.

An appropriate environment for this model to be utilised is a multi-institutional one. Each institution maintains an Authority site and allows “Providers” and “Consumers” to be connected to it. A SeLeNe is thus formulated collaboratively since some services required for local task completion may be hosted by “Providers” connected to some remote institution’s Authority.

Services In this model the main Function patterns appearing (see Figure 21) are the “Direct” and “Mediated.” In Table 2 we again provide a proposed grouping of services. Authorities act as entry points and direct participants as to where a service can be found and accessed. They also provide for the registration and maintenance of user profiles, maintenance of registered LO descriptions and syndication. The

A1, A2	**P1, P3**	**P2,P4**	**C1, C2**
Core	Core	Core	Core
Sign-on	Collaboration	Presentation	
Locate	Update	View	
Syndication	ECA		
User Registration			
LO Registration			
Trails/Adaptation			

Table 2: Mediation-based Model Service Placement

services that achieve this functionality are listed below Core for the Authorities column. Remaining services comprise a second group and are provided by participating "Provider" sites. It is not mandatory that the services listed under 'P1, P3' or 'P2, P4' are found together; however, it is important that the Authority services are provided as listed. The table illustrates the minimal requirements so that a SeLeNe with full functionality is formed. "Client" sites are not expected to host any other services beyond Core.

Obviously, as an environment becomes more dynamic it also becomes more difficult to provide strict service placement recommendations. This is mainly due to the fact that sites become less reliable in participating in the system and in providing their services. We assumed in this model that the degree of dynamicity is very low for Authority sites, moderate for "Providers" and high for "Consumers." As is more apparent in the Autonomic model (see Section 3.3.3 below), when building up an e-Learning Network, where to place which services is driven by the requirements of the participants in order to slide the trade-off indicator between high availability and high dynamicity.

Examples for the Mediation-Based Model

(i) A new node joins a SeLeNe

For this model, imagine that the node requesting to join the SeLeNe is 'C1'. The process is very similar to the centralised case because the Sign-on service is available at 'A1' and it may not need to be located first. The basic difference is that 'A1', which is directly known to 'C1', may be unavailable (not the common case) and therefore 'C1' will need to contact Authority 'A2' in order to join the SeLeNe. It is assumed here that provider sites have already been connected to the Authority sites and therefore their services are discoverable and thus if a provider is removed from the SeLeNe, the functionality will still be available through other provider sites.

(ii) A user or group generates a view over the SeLeNe descriptions

Consider in this case site 'C2'. It will first need to locate the Presentation service in order to provide the view parameters. 'A2' is accessed and directs 'C2' to 'P4'. Note that the View service also resides on 'P4' for this example and thus the Presentation service does not need to contact another site. However, unlike the centralised case, multiple Query service instances need to be 'spawned', initiated by the View service. The Query service will be called locally on 'P4' which will request services from the Authority. This is necessary because access to descriptions need to be made transparent to the View services, since descriptions are registered with Authorities (by the consumer sites where the original description bases are stored).

If the view is to be materialised 'P4' calls the Locate service to discover the sites which run the ECA and Update service (that is, site 'P3') to carry out this functionality. This is also an interesting case of illustrating how the multiplicity of services (replication) will work: if it turns out that 'P3' is unavailable, the Locate service at 'A2' will contact 'A1' and direct 'P4' to site 'P1', which also runs the ECA and Update services as shown in Table 2. The actual materialisation will require calls to each individual Access service at the sites involved in the query.

3.3.3 Autonomic

Description In an Autonomic environment, SeLeNe services are highly distributed among participating sites and multiple communications are typically necessary to complete some task. However, although sites do depend on services offered by other sites, the term "Autonomic" is not a misnomer. Autonomy of the sites is attributed to the liberty of entering and leaving the network at will. In addition, the formation of a SeLeNe is based on the requirements of a group of sites to actually create an e-learning network in order to share their LOs. A group of SeLeNe sites may join together to form an ad-hoc e-learning community which may not include all SeLeNe services, but note that in this case it will also not enjoy the complete SeLeNe functionality. For example, it is possible for two sites to interact and exchange information on their local RDF descriptions using only Core services (e.g. one site to query the other) but there is not much to be gained unless both sites share a common ontology and need no further functionality (e.g. collaboration, receive event updates, etc.).

We have identified a number of open issues for P2P requirements for the Grid here and in Deliverable 3, when considering an open and dynamic architecture. Specific solutions are still under development, and standards are expected to be provided by the GGF-P2P working group.

Authorities in an Autonomic model do not differ much from other sites. However, they do have some special characteristics. The idea is taken from real-life learning groups where one person in the group is officially or unofficially assigned the role of group leader: a site that is willing to initiate the formation of a SeLeNe will be named the Authority and it will be contacted to include more sites. Therefore, an Authority will need to host at least the *Sign-on* and *Locate* services. A SeLeNe may contain multiple Authorities which may share participation information. These Authorities may also participate in multiple SeLeNes themselves. In Figure 24 we show a case of two Authorities, A1 and A2, and four other provider sites, P1–P4, that make up a SeLeNe. Again, not all possible connections are shown. Any site may connect to any other site as soon as a required service is located. In addition to Authorities, other sites that provide services may be part of multiple SeLeNes (see Figure 25). Autonomous SeLeNes may communicate via a representative which will of course be an Authority, but it is possible that a network is isolated from all other sites and acts as a private virtual (e-learning) organisation.

In this model, since no one site can be trusted to remain on-line for long, there is no purpose in creating centralised description stores. Each site still stores its LO descriptions locally but it will provide replies on them only when specifically queried, usually as a result of a *Syndication*, *Update* or other service executing at some remote site. Specific P2P-like solutions for querying and organising metadata are referenced in Deliverable 3. Under this model, LO Registration can be utilised, when available, to extend the degree of distribution of one site's local content descriptions. For example, if two Authorities run a Registration service, one can register the descriptions that its neighbouring sites registered to itself initially.

We reiterate that in the formation of an Autonomic SeLeNe a reduced set of available services will result in reduced functionality. Therefore, sites should be equipped with some local functionality that does not require any changes to the services' APIs. Each site is responsible for storing its user profiles using the local storage service and for uploading and updating the profiles whenever the *Trails and Adaptation* service becomes available in order to take advantage of its functionality.The concept of using local SeLeNe-specific agents is attractive. This idea has already been adopted by the ELENA project which utilises agents over, and not within, the P2P network. Agents can automate a number of user tasks such as probing Authorities for Adaptation, Update and ECA services. This is not a far-fetched scenario as most network-edge learners make use of high-bandwidth (almost) always-on connections such as LAN, ADSL and Cable. The Autonomic model is mostly suitable for cases where the creation of ad-hoc collaboration teams is required. An example would be a group of researchers from different universities working on a project proposal, sharing their resources, establishing links between them, structuring and finally updating them as the proposal is being prepared.

Services Due to the dynamic nature and unpredictable availability of services it is quite difficult for a grouping to be decided. Table 3 shows a possible placement policy in which all services are shared by participating sites. Note that this is a minimal service placement scenario and no service (besides Core and Authority services) is duplicated. The sole requirement is that Authorities should at least be hosting *Sign-on* and *Locate* services so that the formation of a SeLeNe can be initiated. A general consideration could be to host related services together at a single site. For example,

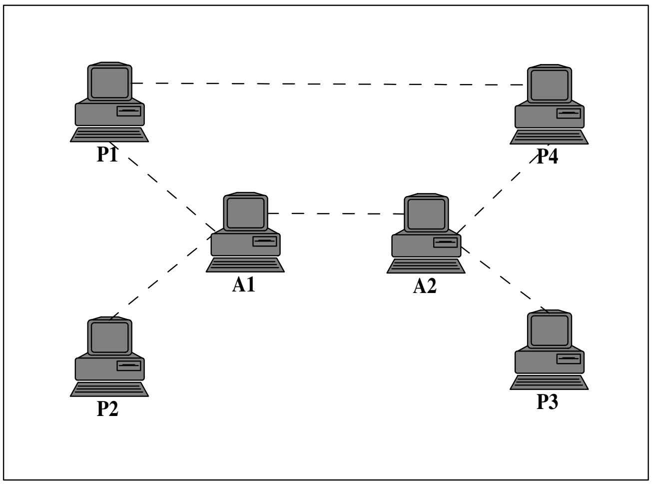

Figure 24: Autonomic Model

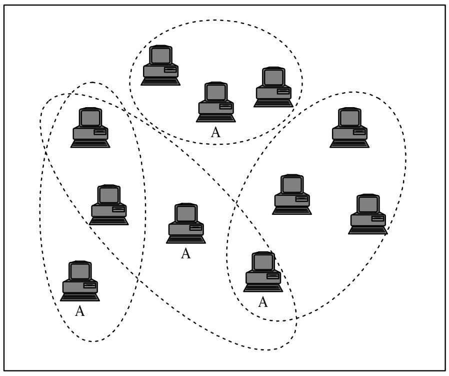

Figure 25: Formation of Multiple Autonomic SeLeNes

although all services are related one way or another, the *User Registration* and *Trails and Adaptation* services are rather more tightly coupled and so it would make sense to host them both at a single site.

In the autonomic model any service pattern (see Figure 21) is possible. "Mediated" will be the prevailing one since services are spread throughout the network, which means that in order to complete a task multiple communications are typically required.

Examples for the Autonomic Model

(i) A new node joins a SeLeNe

For this example we take 'P2' to be the requesting site. Unlike the examples for the centralised and mediation-based models, with the *autonomic* model there are two different possibilities for this use case: either 'P2' is aware of the initiator (Authority) and will

A1	**A2**	**P1**	**P2**	**P3**	**P4**
Core	Core	Core	Core	Core	Core
Sign-on	Sign-on	User Registration	Syndication	View	ECA
Locate	Locate	Trails/Adaptation	Collaboration	Presentation	Update
LO Registration					

Table 3: Autonomic Model Service Placement

request the Sign-on service, or it is connected to (or discovers) a number of neighbouring sites without knowing about the Authority. If the former case holds the process is similar to the previous models. In the latter case, 'P2' will need to "query" its neighbours to discover either the Sign-on or Locate services. Although service discovery is realised by the Locate service, in this case a site is not always aware of an Authority. Service identification is made possible through the core Information service. 'P2' will locate the Sign-on service on 'A1' and will complete its registration along with its offered services.

(ii) A user or group generates a view over the SeLeNe descriptions

We continue using 'P2' for this example, since no service for view definition is available locally for this site. Since one or more Authorities are known, 'P2' will request the Locate service on 'A1' to direct it to a Presentation service at 'P3'. The view parameters are forwarded to the View service, also available at 'P3' (otherwise the Locate service would have been requested by 'P3' from 'A2', to discover the View service). Note that although 'P3' and 'P2' are aware of different Authority sites their communication is direct. As with the mediation-based case, in order for the View service to complete, multiple Query service processes need to be instantiated. It is noted that Table 3 shows a minimal distribution of services. Typically, multiple services are hosted on participating sites, thus providing replication alternatives. Additionally, sites possibly form and participate in multiple different SeLeNes.

3.3.4 Discussion of Comparative Parameters

In this section we compare the three models on a number of parameters, mostly related to their feasibility with respect to the current status of the Grid, as well as the environment for which they are to be set up. We do not consider quantitative measures as the primary purpose of the study is to map the set of proposed services to different e-learning scenarios that require different architectural approaches. These are summarised in Table 4.

Moving our discussion to the comparative parameters, we firstly investigate whether there exist available deployment options and whether current Grid technology suffices. We have included in our report a discussion of what today's Grid technology needs to consider and in which areas there is an opportunity for extending and complementing current functionality. These include the incorporation of Semantic Web technologies, which will contribute greatly to realising the proposed models. At this time a Centralised model seems feasible and can be deployed over Web Services while the management of services can be assisted by Globus. A Mediation-based model is also feasible by extending the Centralised model across multiple organisations. However, in this case methods for metadata distribution will need to be utilised. Obviously, since the Autonomic model combines both P2P and Semantics, it is harder to deploy — this problem is the basis of extensive research taking place towards the combination of these two areas [34, 21, 14, 16].

The second parameter considered is extensibility. A model's extensibility can be separated into three distinct areas: the updating of services, scalability, and flexibility (i.e. the ability of one model to shift to another). Updating and modifying service functionality is obviously easier the more centralised the model is. On the other hand, creating new services and incorporating them into the SeLeNe is simple with the Autonomic model. Therefore, extensibility from this point of view is based on a group need to create new services. Scalability is quite straightforward. A Centralised model is harder to scale than the Mediation-based or Autonomic models but again this is dependent on the learning environment. A single institution which offers a number of courses or an organisation setting up an online training program do not have high requirements to scale their network. On the other hand, Mediation-based and Autonomic models will be employed for a multi-institutional or ad-hoc environment where scalability is an issue. Site dynamicity and reliability (and, as a consequence, service availability) are decisive factors for a model's flexibility in shifting towards another model. This was exactly the reason that we considered multiple architectural scenarios since they fit into different learning environments. Therefore, one model can naturally shift towards another model as conditions such as site dynamicity and organisation (e.g. flat, hierarchical, etc.) change. Consider, for example, an Autonomic network in which specific sites are highly active. These could evolve to brokering and mediation Authorities as in a Mediation-based model.

Metadata distribution is also a basic parameter that should be considered in distinguishing the models. As *distribution of services* and *distribution of data* change, the deployment conditions and assumptions may also change: service deployment depends on input data (i.e. what functionality is appropriate for this kind of input) and data content and structuring must be sufficient for each service to operate upon it (i.e. what input is required to achieve this service's functionality). Thus we address two questions:
(i) How does distribution of data affect services?
'Data' is primarily the resource descriptions stored at each site's repository. Therefore, services affected by the distribution of data (i.e. the description bases) are those that require access to the repositories. We specifically consider the Query service and its functionality and results with respect to distribution. Although the Query service has a specified functionality, how this is implemented at each participating site may vary. A Centralised model implies a centralised repository where a single request is sent and processed (without considering other intermediate requests such as Syndication), whereas in the Autonomic case multiple query requests must be sent to different sites and multiple Query service instances need to be invoked to produce the final result. The specifics of this will also be affected by the degree of completeness and query precision a user needs. A centralised repository will always produce the same, complete results to the same query, assuming no

change to the repository. The Mediation-based model will also produce accurate results as long as all Authorities can be reached. However, the results returned within an Autonomic model cannot be guaranteed always to be complete.

(ii) How does distribution of services affect data?

Although descriptions can be stored at different sites, they are all eventually considered as part of the same learning space created by the descriptions. Thus, since multiple services need to access, exchange and manipulate the data, these data should remain consistent. This requires that each "fragment" be self-contained. That is, different services dealing with a specific part of a repository during some function should be able to identify its origin and "status so far". As distribution of services grows, more control information needs to be exchanged. This might involve overloading the descriptors with this information. In a centralised environment there exists the ability for services to be aware of the repository status at any given time. The mediation-based approach allows for partial knowledge of status for the parts of the repositories that are registered with each Authority. For the Autonomic model, status information needs to be explicitly present within the exchanged data, or must be distributed using additional control messages.

Another comparative parameter is the "dynamicity versus service availability" trade-off. The Centralised model is traditionally the most reliable but also the least dynamic. If the centralised server breaks down, no functionality is possible. On the other hand the Mediation-based model is more dynamic than the centralised as Authorities may continue providing services even if they are connected to no other Authority. However, a Mediation-based model is not fully dynamic and Authorities are assumed to be adequately reliable. The Autonomic model is highly dynamic and availability of services cannot be assured, but this does not mean that this will necessarily have major impact on the achieved functionality since, at the same time, multiple service instances are present in the network.

Choosing among the various models (and the models in-between them) is a classic software engineering problem: Users will state that "I don't know what I want but I will know it's the right one when I see it." However, since we have established an architectural framework by explicitly defining the required services based on user requirements, we may choose the deployment model which is closest to our learning environment. Similar to learning itself, the environment evolves and reaches its final form. It is this process of evolution through information semantics that requires further extensions and complementary standards to Grid technologies to interact with and take advantage of Semantic Web technologies, a task that is already under way at GGF and to which SeLeNe aims to contribute.

4 Exploitation Plans for SeLeNe

Don Peterson
Institute of Education, University of London

	Centralised	**Mediation-Based**	**Autonomic**
Deployment Opportunities	Possible with current technologies. The Semantic Grid.	Available technologies such as JXTA. Add semantic metadata manipulation capabilities to Grid servers.	Harder deployment. Enhancement of the Grid to accommodate P2P support.
Extensibility / Model shifting	Easily extensible but cannot scale services.	Possible to add new services, difficult to maintain smooth intercommunications.	Difficult to change services but flexible to evolve.
Distribution of Metadata	Low	High	High
Load Balancing / Management	Possible	Difficult	Very difficult
Dynamicity	Low	Low or predictable for Authorities, moderate for "clients".	High
Availability	High	Moderate to high	Low to Moderate
Example Application	On-line University Courses ("Blackboard"-style).	Inter-organisation employee-training services.	Ad-hoc collaborative teams (e.g. project consortium collaboration).

Table 4: Summarised Comparison

4.1 Enterprise Knowledge

The field of Knowledge Management arose from a recognition that knowledge may exist somewhere within an organisation and yet be 'implicit' to (or not located by) a particular individual or team which needs it. More specifically it is also recognised as important to support the transfer of procedural 'knowhow' from one employee or team to another, thus avoiding wastage of time and costs. These challenges increase as an organisation becomes larger, and the application of SeLeNes in this context could be fruitful. In this case, a task profile might replace a learner profile, some of the LOs would be internal documents, and the communities generating metadata might be localised or distributed teams or other categories of employee.

If the European Research Area (ERA) can be considered as a type of organisation, a pilot study might be conducted on the issue of procedural knowhow in applying for and conducting EU research projects. Alternatively, a pilot study might be conducted with the cooperation of the "Head of Knowhow" of a large multinational law firm.

4.2 CORDIS and the ERA

Members of a distributed research community need effective methods of identifying relevant work. This challenge is especially acute in relation to work which is very new, or which has not yet received wide recognition; and the challenge magnifies as the community becomes larger or more heterogeneous. Issues of trust and community arise here: an individual researcher may place particular trust in the recommendations of a particular (perhaps distributed) community. A SeLeNe might therefore provide useful service in this context, allowing subcultures within a larger distributed community to provide recommendations of objects and trails of objects, thus supporting both diversity and integration in research. This is potentially relevant to the ERA, to the research output of EU projects, and to the CORDIS website.

A pilot study might be conducted on a Network of Excellence such as Kaleidoscope, or within one of its SIGs.

4.3 Education

In the more exploratory aspects of learning, learners need to find LOs which are relevant to their work, and in doing this they often rely on recommendations from individuals and communities. For example, a relationship such as "X $is-a-good-introduction-to$ Y" is important, and different individuals may naturally rely on different communities for such recommendations. This is perhaps especially relevant to higher and lifelong learning, and to contexts in which the range of potentially relevant resources is large and distributed. That SeLeNes might provide useful support in this context has been assumed throughout the project.

A pilot study might be conducted as an adjunct to a large empirical project in e-learning.

References

[1] Access Grid project. Available at http://www.accessgrid.org/.

[2] CANDLE: Collaborative And Network Distributed Learning Environment. Available at http://www.candle.eu.org/.

[3] Combechem project. Available at http://www.combechem.org/.

[4] Crossgrid project. Available at http://www.crossgrid.org/.

[5] Datagrid project. Available at http://www.eu-datagrid.org/.

[6] EASEL: Educator Access to Services in the Electronic Landscape. Available at http://litc.sbu.ac.uk/easel/.

[7] Educanext (universal) project. Available at http://www.educanext.org/.

[8] Edutella project. Available at http://edutella.jxta.org/.

[9] Elena project. Available at http://www.elena-project.org/.

[10] European Grid of Solar Observations (EGSO). Available at http://www.mssl.ucl.ac.uk/grid/egso/.

[11] GESTALT: Getting Educational Systems Talking Across Leading-edge Technologies. Available at http://www.fdgroup.co.uk/gestalt/about.html.

[12] Globus toolkit. Available at http://www.globus.org/.

[13] Information services in the globus toolkit. Available at http://www.globus.org/mds/.

[14] The OGSA P2P group. Available at http://www.gridforum.org/4_GP/ogsap2p.htm.

[15] Open Grid Services Infrastructure (OGSI) — draft. Available at http://www.gridforum.org/ogsi-wg/drafts/draft-ggf-ogsi-gridservice-23_2003-02-17.pdf.

[16] The SWAP system. Available at http://swap.semanticweb.org/public/index.htm.

[17] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proc. 10th IEEE Symposium on High Performance Distributed Computing*, 2001.

[18] M. Fowler. *UML Distilled - Third Edition.* Addison-Wesley, 2004.

[19] C. Goble and D. De Roure. Semantic Web and Grid computing (preprint), September 2002. Available at `http://www.semanticgrid.org/documents`.

[20] C. Goble, C. Wroe, and R. Stevens. The myGrid project: services, architecture and demonstrator. In *Proceedings UK e-Science, All Hands Meeting*, 2003.

[21] OGSA P2P Group. Taxonomy of peer to peer architectures (version 09.5), 2001. Available at `http://batalion.ucsd.edu/ggf/P2P-Taxonomy-v095.pdf`.

[22] K. Keenoy, M. Levene, and D. Peterson. SeLeNe Deliverable 4.2: Personalisation and trails in Self e-Learning Networks, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[23] K. Keenoy, G. Papamarkos, A. Poulovassilis, M. Levene, D. Peterson, P.T. Wood, and G. Loizou. SeLeNe Deliverable 2.2: Self e-Learning Networks – functionality, user requirements and exploitation scenarios, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[24] H. Lican, W. Zhaohui, and P. Yunhe. A scalable and effective architecture for Grid Services' discovery. In *Proceedings of the First Workshop on Semantics in Peer-to-Peer and Grid Computing, in conjunction with the Twelfth International World Wide Web Conference*, 2003.

[25] A. Lser, W. Nejdl, M. Wolpers, and W. Siberski. Information integration in schema-based peer-to-peer networks. In *Conference on Advanced Information Systems Engineering (CAiSE)*, pages 258–272, 2003.

[26] A. Magkanaraki, V. Christophides, and D. Plexousakis. SeLeNe Deliverable 4.3: Views and structured querying in Self e-Learning Networks, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[27] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[28] G. Papamarkos, A. Poulovassilis, and P. T. Wood. Event-condition action rules languages for the semantic web. In *1st Workshop on Semantic Web and Databases, Berlin*, 2003.

[29] G. Papamarkos, A. Poulovassilis, and P. T. Wood. SeLeNe Deliverable 4.4: ECA rule languages for active Self e-Learning Networks, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[30] P. Rigaux and N. Spyratos. SeLeNe Deliverable 4.1: Generation and syndication of learning object metadata, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[31] D. De Roure, N. R. Jennings, and N. R. Shadbolt. The Semantic Grid: A future eScience infrastructure. In *Proceedings UK e-Science*, National e-Science Centre. University of Edinburgh, 2002.

[32] G. Samaras, K. Karenos, and E. Christodoulou. SeLeNe Deliverable 3: A Grid Service framework for Self e-Learning Networks, 2003. Available from `http://www.dcs.bbk.ac.uk/selene/reports/`.

[33] H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-based resource matching in the Grid, the Grid meets the Semantic Web. In *Proceedings of the First Workshop on Semantics in Peer-to-Peer and Grid Computing, in conjunction with the Twelfth International World Wide Web Conference*, 2003.

[34] S. Ternier, E. Duval, and F. Neven. Using a P2P architecture to provide interoperability between learning objects. In *EdMedia World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2003.