



Creating a Smart Space for Learning

Artefacts and Service Network Architecture Specification Report version 2

Deliverable Number:	D2.2
Contractual Date of Delivery:	1 st July 2003
Actual Date of Delivery:	1 st July 2003
Work Package(s) Contributing:	WP2
Nature of the Deliverable:	Report
Status:	Final
Security (Distribution Level):	Public
Editor(s):	P. Dolog, W. Nejd, D. Olmedilla (UHANN)

The ELENA Consortium consists of:

CSI-Center for Social Innovation	Financial Coordinator	Austria
Wirtschaftsuniversität Wien (WUW)	Scientific Coordinator	Austria
Infonova (INF)	Contractor	Austria
National Centre for Scientific Research "Demokritos" (NCSR)	Contractor	Greece
Allweb Solutions S.A. (ALW)	Assistant Contractor	Greece
Institute "Jozef Stefan" (IJS)	Contractor	Slovenia
Universidad Politécnica de Madrid (UPM)	Contractor	Spain
Landssimi Island (LSI)	Contractor	Iceland
Universität Hannover (UHA)	Contractor	Germany
CDI Deutsche Private Akademie für Wirtschaft (CDI)	Contractor	Germany
imc information multimedia communication (IMC)	Contractor	Germany

Amendment History

Version	Date	Editors(s)	Modification
1.0	2 nd December 2002	P. Dolog, W. Nejdl, D. Olmedilla (UHANN)	Artefacts and Service networks specification focusing integration with LMS
1.3	4 th May 2003	P. Dolog, W. Nejdl, D. Olmedilla (UHANN)	Super-Peer architecture added – focus on advanced P2P exchange using Super Peers; Architecture for personalisation services added
1.9	24 th May 2003	P. Dolog, W. Nejdl, D. Olmedilla (UHANN)	Implementation and integration details added (current state of ELENA architecture + realised scenarios for integration). TRIPLE provider and language description added
2.0	24 th June 2003	P. Dolog, W. Nejdl, D. Olmedilla (UHANN)	Database Federation approach, PLA description and Educational Nodes update.

Contributors

Name	Institution
Stefan Brantner	Infonova
Ingo Brukhorst	UHANN
Peter Dolog	UHANN

Wolfgang NejdI	UHANN
Daniel Olmedilla	UHANN
Reinhard Pawis	CDI
Bettina Preßmann	IMC
Bernd Simon	WUW
Gerald Thiel	CDI
Martin Wolpers	UHANN
Menelaos Yoanidis	AllWeb
Juan Quemada	UPM
Joaquín Salvachúa	UPM
Michael Sintek	DFKI

Table of Contents

Executive summary.....	6
1. Introduction	7
2. Artefacts	8
3. Architecture	9
3.1 System Components.....	9
3.2 ELENA network.....	10
3.3 Exchange Process Supported.....	11
4. Edutella Framework.....	12
4.1 The JXTA P2P Framework	13
4.2 Edutella Services	13
4.2.1 Query Service	14
4.2.2 Edutella Replication	14
4.2.3 Edutella Mapping, Mediation, Clustering	14
4.3 Schema-based P2P network.....	15
4.3.1 Schema-based routing in P2P networks	15
4.3.2 Routing Super-Peer-Peer Queries and Responses.....	16
4.4 Providers.....	22
4.5 RDF-QEL-i Language Levels	23
4.6 TRIPLE Overview	24
4.6.1 Namespaces and Resources	24
4.6.2 Statements and Molecules	25
4.6.3 Models	25
4.6.4 Logical Formulae.....	26
4.6.5 Clauses and Blocks.....	26
4.7 A Model for the Federation of Communities and Metaproviders.....	26
4.7.1 Agreements among communities	27
4.7.2 Implementation.....	28
4.7.3 Example.....	28
4.8 Focusing Personalisation Services.....	29
5. Scenarios for Integration	32
6. Interfaces	35
7. Network Nodes and Artefacts.....	37
7.1 Educational nodes and artefacts.....	37
7.1.1 EducaNext	37
7.1.2 ULI Campus (Universitärer Lehrverbund Informatik).....	61

7.2	Other Nodes	68
7.2.1	Personal Learning Assistant	68
7.2.2	Java client for manually operating the Synchronization interface	70
7.2.3	Edutella consumer	70

Executive summary

Smart Spaces for Learning are the next step towards the semantic web. In the ELENA project we try to design and develop such smart learning spaces based on a peer-to-peer approach. According to this approach, the smart learning space can be seen as an open network of peers which collaborate on finding appropriate learning services or resources for specific persons.

In this document we address architecture descriptions of the service network and several points of view for artefacts in that network. We address software components, metadata and resource points of view to artefacts. The service network architecture is based on Edutella framework. The advanced P2P exchange based on introducing Super Peers is discussed. The network is discussed from the personalisation services point of view as well. We give several possibilities (scenarios) for integration. Interfaces as the main means for integration are discussed. For each considered educational node the metadata artefacts and resource artefacts are described. We describe the integration scenario implemented for each educational node.

1. Introduction

In order to achieve the main objective of the ELENA project, namely to demonstrate the feasibility of smart learning spaces, a sound technical infrastructure is needed. The smart learning space will be constituted from a distributed learning repository of learning resources and services. The learning repository will be accessed through interfaces of the learning management systems or by querying Edutella [2]. The learning management systems will be connected in the smart learning space through the Edutella infrastructure. To be able to search for specific services in a smart learning space the query service should be provided. It means that by this query service we will be able to retrieve service, which is currently available and accessible in the smart learning space and can be constituted from resources which are distributed. The resources will be maintained by existing learning management systems. The access to these resources and their descriptions should be provided by means of the interface. Through this interface the learning management system can be connected to the smart learning space where it will serve as a provider.

In this document we are trying to address the possibilities of integration of Edutella infrastructure with current educational nodes (learning management systems and/or mediators). The Edutella infrastructure serves as the mediator for querying distributed learning resources or in the future learning service providers.

We address schema based routing approach for improving metadata artefact discovery. The schema based routing is provided by introduced Super Peers.

We address personalisation services in the context of service network as well. The personalisation services will allow us to personalize learning services execution.

We have employed a spiral lifecycle approach within the ELENA project. It means, that this document reflects current state of the art and current design decisions. The document is subject to changes.

The rest of the document is structured as follows. Section 2 is devoted to the discussion about artefacts. Section 3 describes the architecture of the service network as well as the current state of implementation. Section 4 describes the background of Edutella – JXTA framework, Educational context and Edutella services. This section also provides the discussion and architectural drawbacks based on schema based routing allowing advanced P2P exchange. The personalisation services based on reasoning capabilities of peers are discussed as well. A federation model for P2P is discussed as well. The query language supported by Edutella and currently available providers are also discussed in this section. Section 5 provides three high level scenarios for integration of Edutella and educational nodes. Section 6 provides interfaces descriptions for providing and accessing metadata. Section 7 gives an overview of the network nodes in our project as well as the artefacts provided by considered educational nodes.

2. Artefacts

There are four main views of artefacts in a smart learning space discussed in this document:

- Software components
- Metadata artefacts
- Resource artefacts
- Systems

Software components realize access and provision of resource and metadata artefacts. These artefacts can be described from two points of view:

- Which technical services they provide (provision, learning, evaluation...)
- Which kind of artefact they provide (resource/content, metadata, educational services)

The first view is discussed in the “Architecture” section. The second view is discussed in the “Interfaces” section.

The metadata and resource artefacts are subjects of authoring and are changing. It means that they are related to a specific provider of software component, which supports the authoring and managing of these resources, services and metadata about them. Thus we describe these artefacts in the “Educational nodes and Artefacts” section.

The systems’ point of view considers which systems as metadata and resource providers are connected to the ELENA network. The current state of existing systems as artefacts in the network is also discussed in the “Network nodes and Artefacts” section.

3. Architecture

3.1 System Components

Figure 3.1 depicts the various components of a smart learning space. In a smart learning space, educational service providers are connected within the learning management network which is based on Edutella [2]. Edutella is a peer-to-peer (P2P) technology that aims at connecting highly heterogeneous educational peers with different types of repositories, query languages and different kinds of metadata schemas. Each Edutella peer is capable of performing a number of basic services such as querying, replication and mapping. Educational services providers either connect directly to the network as Edutella peers (Educational Nodes in our context) or use other Edutella peers (e.g. the Universal Brokerage Platform for Learning Resources) as a gateway to announce their services in the network.

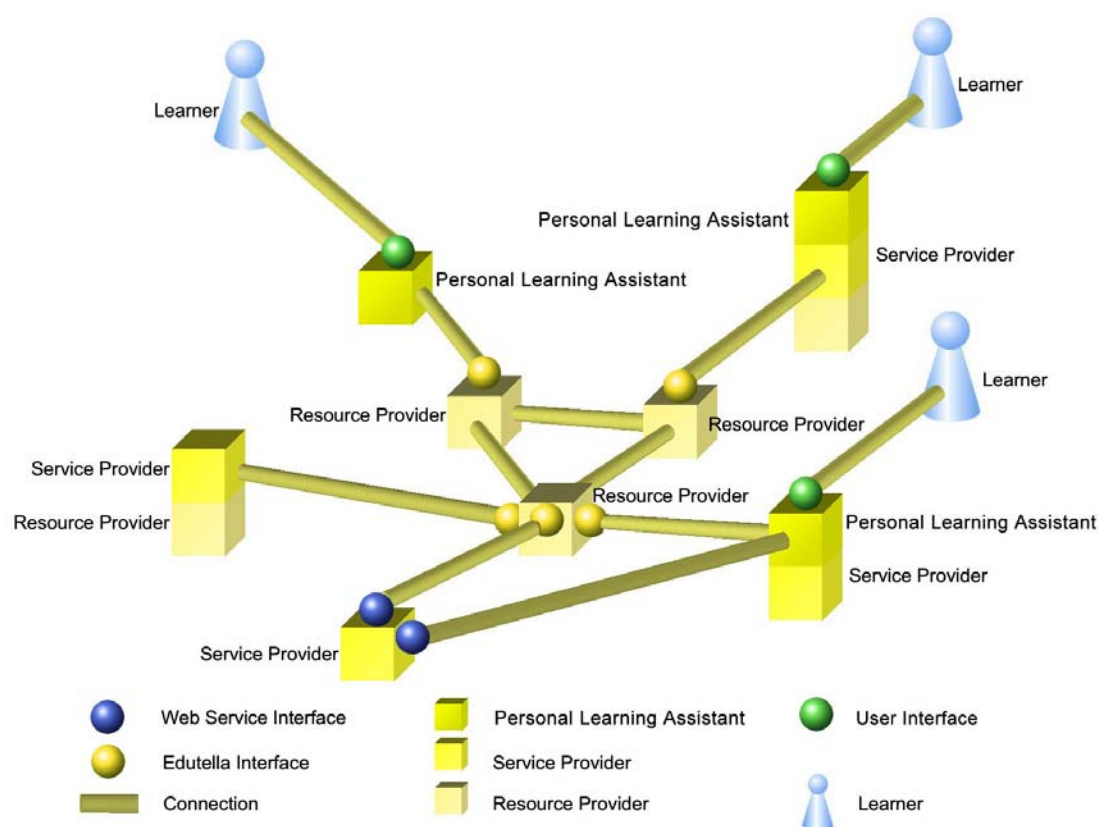


Figure 3.1: System Components of Smart Learning Spaces

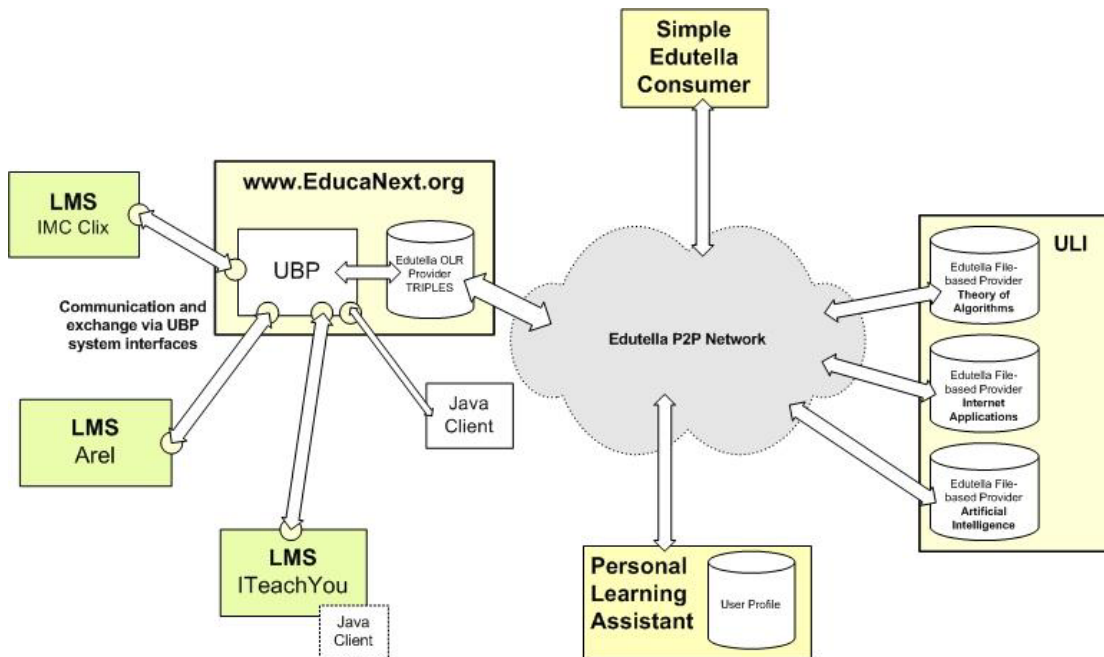
Educational service provider need not to provide educational services in a fully electronic manner. For example, some educational service providers just list a number of courses, whose availability is spread through the learning management network like a static web page but using XML and RDF instead of plain HTML for describing their courses. Others might provide a fully web-based training application and a web-

service enabled interface, which allows automatic learner registration and reporting of learning achievements. Special rating service providers facilitate rating-like annotations of educational service descriptions.

On top of the learning management network Personal Learning Assistants (PLAs) interact with the connected Educational Nodes in order to query for suitable learning services. PLAs take advantage of the learner profile in order to augment queries and personalize query results. They recommend learning services based on the profile via push technology and have rules implemented, which allow them to automatically perform processes such as course registration.

3.2 ELENA network

This section describes the setup of the artefacts and service network as it is currently implemented. Elena runs its own Edutella network.



As illustrated in the above figure the service network consists of an instance of the UNIVERSAL Brokerage Platform (UBP) connected via the Edutella Peer-To-Peer network and an instance of ULI course providers. The instance of the UBP is the EducaNext server installed at WUW. The three Learning Management Systems (LMS) Clix, Arel and ITeachYou are connected to this broker instance. These systems implement the synchronization interface as described in deliverable D1.2 "Learning Management Network Specification". The ULI course providers are running at the KBS site. It is intended to add a new Educational Node based on another instance of UBP in order to fulfil the extra-organizational scenario described in the D1.1 (Basic Service Portfolio). With a simple Edutella consumer the network can be queried for registered learning resources. It is also possible to test the Synchronization interface with a simple Java tool. In the section 7 (Network Nodes and Artefacts) a description of the network nodes follows.

3.3 Exchange Process Supported

Because of the extensive use of resources, companies do not offer the services for free. Hence, an exchange transaction comprising provision, offer placement, announcement, booking, and settlement of educational services needs to be supported by a mediator of educational services. Figure 3.2 illustrates this exchange process.

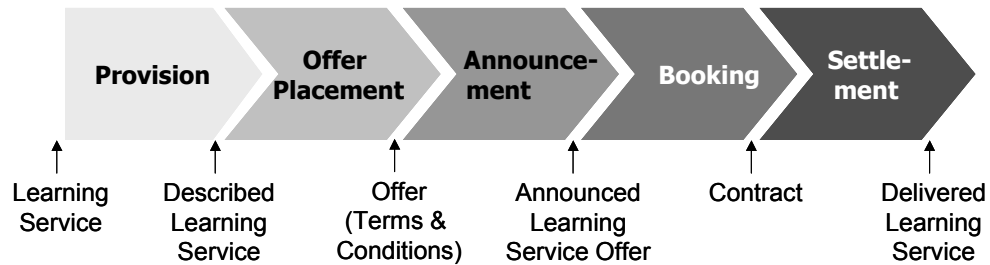


Figure 3.2: Exchange Process of Educational Services

4. *Edutella Framework*

Every Educational Node (e.g. universities) usually already has a large pool of educational resources distributed over its institutions. These are under control of the single entities or individuals, and it is unlikely that these entities will give up their control, which explains why all approaches for the distribution of educational media based on central repositories have failed so far. Furthermore, setting up and maintaining central servers is costly. The costs are hardly justifiable, since a server distributing educational material would not directly benefit the sponsoring university. We believe that, in order to really facilitate the exchange of educational media, approaches based on metadata-enhanced peer-to-peer (P2P) networks are necessary.

In a typical P2P-based e-learning scenario, each Educational Node acts not only as a content provider but also as a content consumer, including local annotation of resources produced at other sites. As content provider in a P2P network they will not lose their control over their learning resources but still provide them for use within the network. As a content consumer both, providers and learners, benefit from having access not only to a local repository, but to a whole network, using queries over the metadata distributed within the network to retrieve required resources.

P2P networks have already been quite successful for exchanging data in heterogeneous environments, and have been brought into focus with services like Napster and Gnutella, providing access to distributed resources like MP3 coded audio data. However, pure Napster and Gnutella like approaches are not suitable for the exchange of educational media. For example, the metadata in Gnutella is limited to a file name and a path. While this might work for files with titles like “Madonna - Like a Virgin”, it certainly does not work for “Introduction to Algebra - Lecture 23”. Furthermore, these special purpose services lead to fragmented communities, which use special purpose clients to access their service.

The educational domain is in need of a much richer metadata markup of resources, a markup that is often highly domain and resource type specific. In order to facilitate interoperability and reusability of educational resources, we need to build a system supporting a wide range of such resources. This places high demands on the interchange protocols and metadata schemata used in such a system, as well as on the overall technical structure. Also, we do not want to create yet another special purpose solution which is outdated as soon as metadata requirements and definitions change.

Our metadata based peer-to-peer system therefore has to be able to integrate heterogeneous peers (using different repositories, query languages and functionalities) as well as different kinds of metadata schemas. We find common grounds in the essential assumption that all resources maintained in the Edutella network can be described in RDF, and all functionality in the Edutella network is mediated through RDF statements and queries on them. For the local user, the Edutella network transparently provides access to distributed information resources, and different clients/peers can be used to access these resources. Each peer will be required to offer a number of basic services and may offer additional advanced services.

4.1 The JXTA P2P Framework

JXTA is an Open Source project [1, 3] supported and managed by Sun Microsystems. In essence, JXTA is a set of XML based protocols [4] to cover typical P2P functionality. It provides a Java binding offering a layered approach for creating P2P applications (core, services, applications, see Figure 4.1, reproduced from [1]). In addition to remote service access (such as offered by SOAP), JXTA provides additional P2P protocols and services, including peer discovery, peer groups, peer pipes, and peer monitors. Therefore JXTA is a very useful framework for prototyping and developing P2P applications.

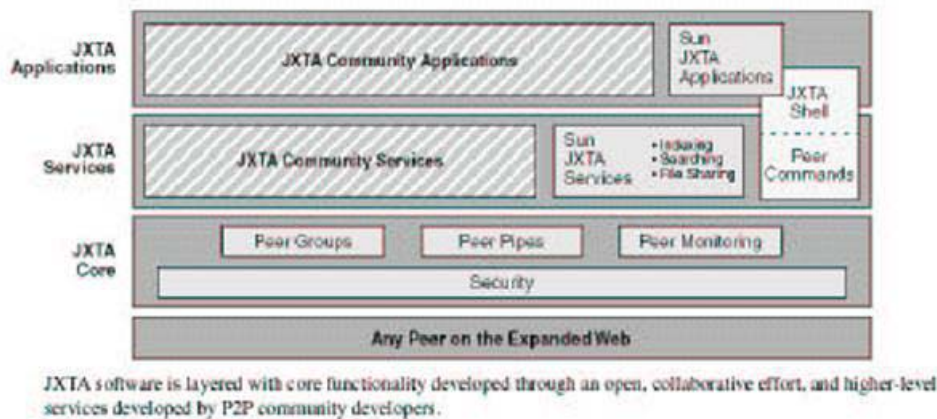


Figure 4.1 JXTA layers

This layered approach fits very nicely into our application scenarios defined for Edutella:

Edutella Services (In the future described in web service languages like DAML-S or WSDL, etc.) complement the Jxta Service Layer, building upon the JXTA Core Layer, and **Edutella Applications/Front-ends** (Services are also part of peers) live on the Application Layer, using the functionality provided by these Edutella services as well as possibly other JXTA services.

On the Edutella Service layer, we define data exchange formats and protocols (how to exchange queries, query results and other metadata between Edutella Peers), as well as APIs for advanced functionality in a library-like manner. Applications like repositories, annotation tools or GUI interfaces connected to and accessing the Edutella network are implemented on the application layer.

4.2 Edutella Services

Edutella connects highly heterogeneous peers (heterogeneous in their uptime, performance, storage size, functionality, number of users etc.). However, each Edutella peer can make its metadata information available as a set of RDF statements. Our goal is to make the distributed nature of the individual RDF peers connected to

the Edutella network completely transparent by specifying and implementing a set of Edutella services. Each peer will be characterized by the set of services it offers.

4.2.1 Query Service

Peers register the queries they may be asked through the query service (i.e., by specifying supported metadata schemas (e.g., “this peer provides metadata according to the LOM 6.1 or DCMI standards”) or by specifying individual properties or even values for these properties (e.g., “this peer provides metadata of the form `dc_title(X,Y)`” or “this peer provides metadata of the form `dc_title(X,'Artificial Intelligence')`”). Queries are sent through the Edutella network to the subset of peers who have registered with the service to be interested in this kind of query. The resulting RDF statements are sent back to the requesting peer.

4.2.2 Edutella Replication

This service is complementing local storage by replicating data in additional peers to achieve data persistence / availability and workload balancing while maintaining data integrity and consistency. Since Edutella is mainly concerned with metadata, replication of metadata is our initial focus. Replication of data might be an additional possibility (though this complicates synchronization of updates). This service is not implemented yet.

4.2.3 Edutella Mapping, Mediation, Clustering

While groups of peers will usually agree on using a common schema (e.g., SCORM or IMS/LOM for educational resources), extensions or variations might be needed in some locations. The Edutella Mapping service will be able to manage mappings between different schemata and use these mappings to translate queries over one schema X to queries over another schema Y. Mapping services will also provide interoperation between RDF- and XML-based repositories. Mediation services actively mediate access between different services, clustering services use semantic information to set up semantic routing and semantic clusters. These services are not implemented yet.

4.3 Schema-based P2P network

In this section, the exchange of information of Learning Object (LO) between Learning Management Systems (LMS) is explained. As normally each LMS has his own LO structure and information (metadata) is necessary to find a common way to understand each LMS to each other. At this point, a new type of P2P networks will be used: schema-based P2P network. Thus, the use of super-peer based topologies [12] for these network provide perfect support for inhomogeneous schema-based networks, which support different metadata schemas and ontologies. Furthermore, they are able to support sophisticated routing and clustering strategies based on the metadata schemas, attributes and ontologies used. Especially helpful in this context is the RDF functionality to uniquely identify schemas, attributes and ontologies. The resulting routing indices can be built using dynamic frequency algorithms and support local mediation and transformation rules.

4.3.1 Schema-based routing in P2P networks

To take the semantic heterogeneity of schema-based P2P networks into account, it will be needed to have a super-peer topology for these networks and the use of indices at these super-peers to address scalability requirements. The super-peer network constitutes the "back-bone" of the P2P network which takes care of message routing and integration / mediation of metadata.

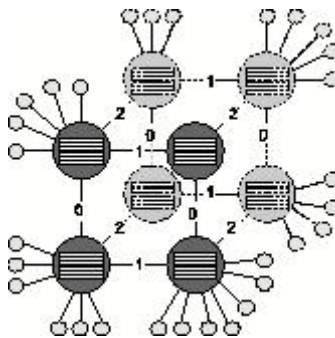


Figure 4.2: peers connected to the super-peer "backbone"

Scaling a P2P network to a large number of super-peers while maintaining certain properties such as low network diameter requires guiding the evolution of the network topology upon peer joins and departures. In Edutella super-peers are organized into a hypercube topology. A new super-peer is able to join the network by asking any other, already integrated super-peer which then carries out the peer integration protocol. This algorithm enables efficient query broadcast and guarantees non-redundant broadcast.

The introduction of super-peers in combination with routing indices reduces the workload of peers significantly by distributing queries only to the appropriate subset of all possible peers.

4.3.2 Routing Super-Peer-Peer Queries and Responses

The first kind of indices needed in super-peers are so-called super-peer/peer routing indices (SP/P-RIs). In these indices each super-peer stores information about metadata usage at each peer. This includes schema information such as schemas or attributes used, as well as possibly conventional indices on attribute values.

On registration the peer provides the super-peer with its metadata information by publishing an advertisement. This advertisement encapsulates a metadata based description of the most significant properties of the peer. As this may involve quite a large amount of metadata, they have been built upon the schema-based approaches which have successfully been used in the context of mediator-based information systems.

To ensure that the indices are always up-to-date, peers notify super-peers when their content changes in ways that trigger an update of the index. For example, if a peer had announced that it uses the Dublin Core schema *dc* during the last connection to its super-peer, but now also uses the Learning Object Metadata schema *lom* to describe resources, it needs to announce this to the super-peer. If a peer leaves the network, all references to this peer are removed from the indices. In contrast to some other approaches, the indices do not contain content elements but peers. At each super-peer, elements used in a query are matched against the SP/P-RIs in order to determine local peers which are able to answer the query. A match means that a peer understands and can answer a specific query, but does not guarantee a non-empty answer set. The indices can contain the information about other peers or super-peers at different granularities: schema identifiers, schema properties, property value ranges, and individual property values.

To illustrate index usage, the following sample query will be used: find lectures in German language from the area of software engineering suitable for undergraduates. In the Semantic Web context this query would probably be formalized using the *dc* schema for document specific properties (e.g. *title*, *creator*, *subject*) and the *lom* schema which provides learning material specific properties, in combination with classification hierarchies (like the ACM Computing Classification System, ACM CCS) in the *subject* field. In line with RDF/XML conventions, properties will be identified by their name and their schema (expressed by a namespace): "*dc:subject*" therefore denotes the property "subject" of the DC schema. So, written in a more formal manner, the query becomes:

Find any resource where the property *dc:subject* is equal to *ccs:softwareengineering*, *dc:language* is equal to "de" and *lom:context* is equal to "undergrad".

The next table shows the values requested in the query at the different granularities; e.g. the query asks for DC and LOM at the schema level, while it requests a *lom:context* value of "undergrad" at the property value level, etc.

Granularity		Query	
Schema		dc, lom	
Property		dc:subject, dc:language, lom:context	
Property	Value	dc:subject	ccs:sw-engineering
Range			
Property Value		lom:context dc:language	"undergrad" "de"

Contents of the sample query at different granularities

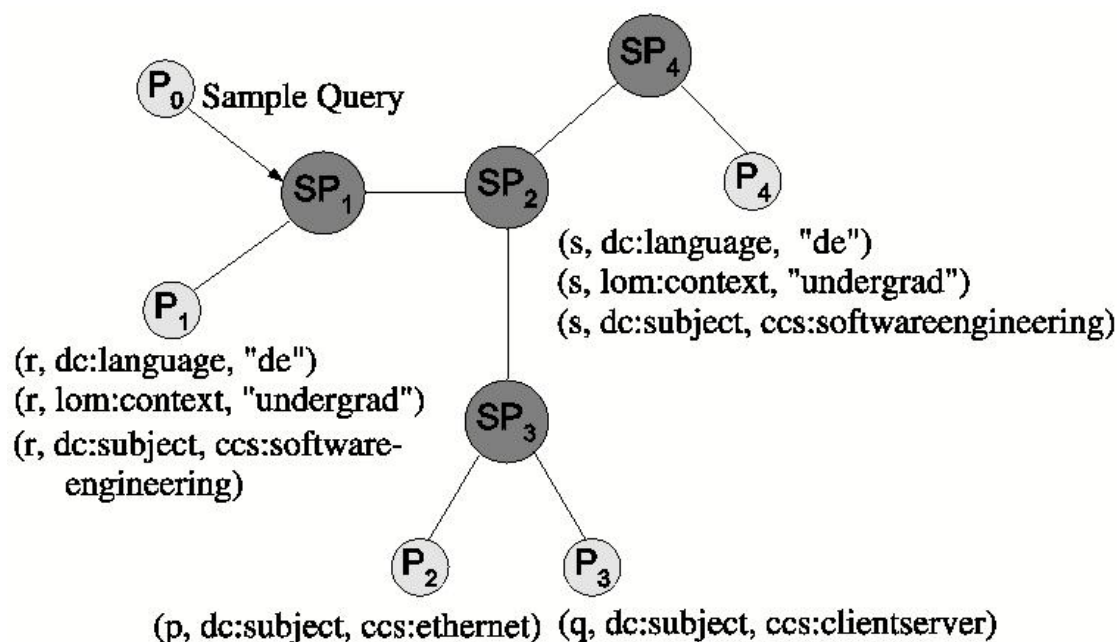


Figure 4.3: routing example network

In order to further clarify things the scenario shown in figure 4.3 is considered. In this network, various resources are described on different peers, which in turn are attached to super-peers.

Peer P0 sends the sample query mentioned above to its super-peer SP1. In our example, this query could be answered by the peers P1 and P4, attached to SP1 and SP4, respectively. These contain metadata about resources r and s which match the query.

The following paragraphs will explain how the routing indices at the different granularities facilitate routing the query to the right peers.

Schema Index

We assume that different peers will support different schemas and that these schemas can be uniquely identified (e.g. the dc and lom namespaces are uniquely identified by an URI). The routing index contains the schema identifier as well as the peers supporting this schema. Figure 4.4 shows a sample of such an index. Queries are forwarded only to peers which support the schemas used in the query. Super-peer SP1

will forward the sample query to attached peers which use DC and LOM to annotate resources (peer P1 in Figure 4.4. Mediation between different schemas will be supported at the super-peer level.

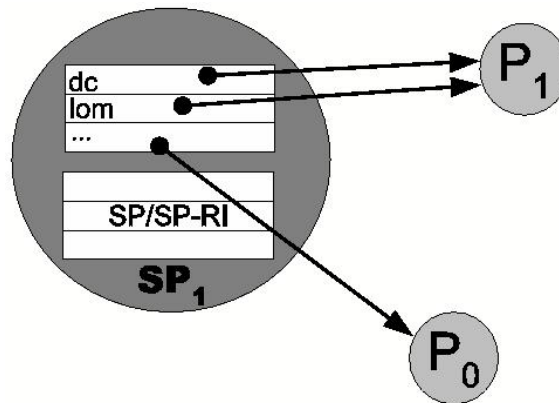


Figure 4.4: super-peer/peer routing index

Property/Sets of Properties Index:

Peers might choose to use only parts of (one or more) schemas, i.e. certain properties, to describe their content. While this is unusual in conventional database systems, it is more often used for data stores using semi-structured data, and very common for RDF-based systems. In this kind of index, super-peers use the properties (uniquely identified by namespace/schema ID plus property name) or sets of properties to describe their peers. Our sample query will be sent to peers using at least `dc:subject`, `dc:language` and `lom:context` (e.g. `SP1` will send the query to `P1`, as `P1` contains all of these properties). Sets of properties can be useful to characterize queries (i.e. a "sets-of-properties index" to characterize and route the most common queries might be used).

Property Value Range Index:

For properties which contain values from a predefined hierarchical vocabulary an index which specifies taxonomies or part of a taxonomy for properties can be used. This is a common case in Edutella, because in the context of the semantic web quite a few applications use standard vocabularies or ontologies. In this example, peers could be characterized by their possible values in the `dc:subject` field, and the query would not be forwarded to peers managing "`ccs:networks`" or "`ccs:artificial_intelligence`" content (as these sub-hierarchies are disjoint from the "`ccs:software_engineering`" sub-hierarchy), and will not be forwarded to peers which use the MeSH vocabulary (because these peers manage medical content).

Note that the subsumption hierarchy in a taxonomy such as ACM CCS can be used to aggregate routing information in order to reduce index size.

Property Value Index:

For some properties it may also be advantageous to create value indices to reduce network traffic. This case is identical to a classical database index with the exception that the index entries do not refer to the resource, but the peer providing it. This index

contains only properties that are used very often compared to the rest of the data stored at the peers.

In the example, this is used to index string valued properties such as `dc:language` or `lom:context`.

4.3.2.1 Routing among Super-Peers based on Routing Indices

As with peers, broadcasting queries to all super-peers should be avoided. To achieve this goal super-peer/super-peer routing indices are introduced to route among the super-peers. These SP/SP indices are essentially extracts and summaries (possibly also approximations thereof) from all local SP/P indices. They contain the same kind of information as SP/P indices, but refer to the (direct) neighbors of a super-peer (as shown in Figure 4.5). Queries are forwarded to super-peer neighbors based on the SP/SP indices, and sent to connected peers based on the SP/P indices.

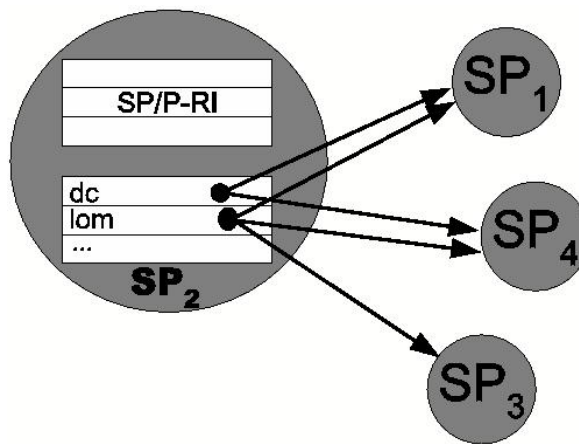


Figure 4.5: Super-peer/super-peer routing index

The next table gives a full example of the SP/SP routing index of SP_2 at the different granularities. For example, SP_2 knows at the schema level that all of its neighbors (SP_1 , SP_3 , SP_4) use the DC name-space, but only SP_1 and SP_4 contain information described in the LOM schema. Thus, the sample query will not be routed to SP_3 , as it requires both DC and LOM.

The same applies for the other levels of granularity. A special case is the Property Value Range level; note that `ccs:networks` is a common super concept of `ccs:ethernet` and `ccs:clientserver` in the ACM CCS taxonomy. Making use of the topic hierarchy, the routing index can contain aggregate information like this in order to reduce index size.

Granularity	Index of SP_2
-------------	-----------------

Schema	dc lom	SP ₁ , SP ₃ , SP ₄ SP ₁ , SP ₄
Property	dc:subject dc:language lom:context	SP ₁ , SP ₃ , SP ₄ SP ₁ , SP ₄ SP ₁ , SP ₄
Property Value Range	dc:subject dc:subject	ccs:networks ccs:software- engineering SP ₃ SP ₁ , SP ₄
Property Value	lom:context dc:language	“undergrad” “de” SP ₁ , SP ₄ SP ₁ , SP ₄

SP/SP index of SP₂ at different granularities

Update of SP/SP indices is based on the registration (or update) messages from connected peers. For the moment it is assumed that a peer can connect to an arbitrary super-peer and define the index update procedure as follows:

1. When a new peer registers with a super-peer, it announces the necessary schema (and possibly content) information to the super-peer.
2. The super-peer matches this information against the entries in its SP/P index.
3. If new elements have to be added in order to include the peer into the SP/P index, the super-peer broadcasts an announcement of the new peer to the super-peer network.
4. The other super-peers update their SP/SP indices accordingly.

Although such a broadcast is not optimal, it is not too costly either. First, the number of super-peers is much less than the number of all peers. Second, if peers join the super-peer frequently, we can send a summary announcement containing all new elements only in pre-specified intervals instead of sending a separate announcement for each new peer. Third, an announcement is necessary only if the SP/P index changes because of the integration of the new peer. As soon as the super-peer has collected a significant amount of peers, these announcements will rather be an exception. Similarly, indices have to be updated when peers disconnect from their super-peers.

4.3.2.2 Mediation between Different Schemas

As outlined before each peer registers with a super-peer using so called advertisements which contain the metadata schema used at the peer. Since schema-based approaches of model correspondences have been successfully used in the context of mediator-based information systems (MBIS) this approach will be applied also to our super-peer networks. In the example in the table of last section it is assumed that each peer provides only one query and one result schema and both schemas are equal. But in some cases a peer will provide many different query schemas and one result schema. In that case both query and result rules will be published to super peers by each peer in the network as a valid advertisement.

Since a peer will only answer queries corresponding to a rule, a super peer will route only relevant queries to its peers. At the moment it can be distinguished between two relevant roles of peers in the network: information provider role and information consumer role. Since peers acting as consumers only will not be able to respond to any kind of queries, they need not be considered by the super peers when looking for suitable advertisements.

Typically super peers will collect several advertisements related to their peers. If a super peer receives a query it tries to identify relevant advertisements matching the schema of the query. It can be distinguished between the following three cases:

1. A query exactly matches an advertisement of only one potential peer.
2. A query exactly matches advertisements of many peers, using one homogeneous schema (or a set of those).
3. A query could be resolved combining results from many peers using heterogeneous schemas.

For case three more sophisticated methods have to be investigated to transform schemas between different peers (i.e. mediation), integrating different query schemas with each other. In the following transformation rules between different schemas will be discussed, so called correspondences, which have already been used in MBIS.

In contrast to MBIS where correspondences are used as rules to translate between global and local schemas, in super peer networks it can be typically assumed only translations between different local schemas. MBIS-based correspondences will be used as rules to describe such translations, and use property names as arguments in query literals for a concise notation.

In the following example the administrator of the super peer defines a query schema `lectures(lecture:identifier, lecture:language, lecture:subject, lecture:educationalcontext)` which will return documents identified by its URL. First correspondences between attributes of the peer schema and the corresponding attributes the lecture schemas will be defined:

1. `lectures:Identifier = dc:title`
`lectures:language=dc:lang`
`lectures:subject=dc:subject`
2. `lectures:Identifier = lom:general.identifier`
`lectures:language=lom:general.language`
`lectures:context=lom:educational.context`

Using the above mentioned correspondences we can now create views on the peer specific schemas:

1. `lecturesViewDC(lectures:Identifier,lectures:language,lectures:subject)`
`← DC(dc:title, dc:lang, dc:subject)`
2. `lecturesViewLOM(lectures:Identifier, lectures:language, lectures:context)`
`← LOM(lom:general.identifier,lom:general.language,`
`lom:educational.context)`

Then we can describe, which attributes of the super peers lectures schema could be answered by the local peer schemas:

1. lectures(lectures:identifier,lectures:language,lectures:subject,-)
← lecturesViewDC(lectures:Identifier,lectures:Language,lectures:subject)
2. lectures(lecture:identifier,lecture:language,-, lecture:context)
← lecturesViewLOM(lectures:Identifier,lectures:Language,lectures:context)

Combining all correspondences then results in two main schema correspondences bridging the heterogeneity between the peers P1 and P2.

Peer1:Correspondence1:

```
lectures(lectures:identifier,lectures:language,-,lectures:educationalcontext)
← v(lectures:Identifier,lectures:language,lectures:context)
← LOM(lom:general.identifier,lom:general.language,\\lom:educational.context)
```

Peer2:Correspondence2:

```
lectures(lectures:identifier,lectures:language,\\lectures:subject,-)
← v(lectures:Identifier,lectures:language,lectures:subject)
← DC(dc:title,dc:subject,dc:lang)
```

A super peer will store relations between correspondences and peers in his indices. When a super peer receives a query lecture (lecture:identifier, lecture:language, lecture:subject, lecture:educationalcontext) the super peer identifies P1:Correspondence1 and P2:Correspondence2 as a combination of relevant correspondences that are semantically included in the user query and is able to compute correct results. The query will then be forwarded to the peers Peer 1 and Peer 2. Afterwards, the results have to be collected and combined by the super peer.

4.4 Providers

Following providers are available in Edutella:

- File-based metadata
- Relational database which supports nested queries
- ConceptBase
- Sesame
- TRIPLE

The file-based provider provides access to RDF files. This provider is able to handle several files. It constructs one RDF model in main memory from files. It is implemented by the JENA semantic web toolkit (see <http://www.hpl.hp.com/semweb/jena-top.html> for details).

A relational database provider provides access to metadata, which are stored in a relational database. The metadata have to be stored in one table. If they are stored in multiple tables, a view of RDF triples is needed. The provider assumes that the database supports nested queries.

A ConceptBase provider provides access to ConceptBase. ConceptBase is a multi-user deductive object manager mainly intended for conceptual modelling and coordination in design environments. The system implements O-Telos, a dialect of Telos which amalgamates properties of deductive and object-oriented languages (see <http://www-i5.informatik.rwth-aachen.de/CBdoc/cbflyer.html> for details).

A Sesame provider provides access to Sesame. Sesame is an RDF Schema-based Repository and Querying facility. As a file-based provider was not enough to manage large amounts of information and Relational Database provider is not optimal for RDF storage this new provider has been developed. Sesame's goal is to provide a stable, efficient and scalable middleware platform for storing, retrieving, manipulating and managing ontologies and metadata stored in RDF, RDF Schema and more expressive languages like OWL. With this new provider performance has been considerably improved.

A TRIPLE provider will provide possibilities for having peers which has reasoning capabilities. TRIPLE is a language for reasoning on the web. TRIPLE provider is under implementation. You can find an overview of TRIPLE in section 4.6.

Edutella providers accept RDF based queries in RDF-QEL-3. Until now, there is only a library of functions, which can be used in an external programming environment to access the functionality of an Edutella peer.

4.5 RDF-QEL-i Language Levels

In the definition of the Edutella query exchange language, several important design criteria have been formulated:

- **Standard Semantics** of query exchange language, as well as a sound RDF serialization. Simple and standard semantics of the query exchange language is important, as transformations to and from this language have to be performed within the Edutella peer wrappers, which have to preserve the semantics of the query in the original query language. Additionally, a sound encoding of the queries in RDF to be shipped around between Edutella peers has to be provided.
- **Expressiveness** of the language. We want to interface with both simple graph based query engines as well as SQL query engines and even with inference

engines. It is important that the language allows expressing simple queries in a form that simple query providers can directly use, while allowing for advanced peers to fully use their expressiveness.

- **Adaptability** to different formalisms. The query language has to be neutral to different representation semantics, it should be able to use any predicates with predefined semantics (like *rdfs:subclassOf*), but not have their semantics built in, in order to be applicable to different semantic formalisms used in the Edutella peers. It should be as easily connected to simple RDFS repositories as to relational databases or object-relation ones, and inference systems, which all have different base semantics and capabilities.
- **Transformability** of the query language. The basic query exchange language model must be easy to translate into many different query languages (both for importing and exporting), allowing easy implementation of Edutella peer wrappers.

Edutella follows a layered approach for defining the query exchange language. Currently we have defined language levels RDF-QEL-1, -2, -3, -4 and -5, differing in expressivity. The simplest language (RDF-QEL-1) can be expressed as unreified RDF graph, the more complex ones are more expressive than RDF itself and therefore have to be expressed using reified RDF statements. All language levels can be represented through the same internal data model [2] (see Appendix A for more information).

Currently, Edutella providers support RDF-QEL-3 language.

4.6 TRIPLE Overview

TRIPLE [16] is a rule language for the Semantic Web which is based on Horn logic and borrows many basic features from F-Logic [17] but is especially designed for querying and transforming RDF models. TRIPLE can be viewed as a successor of SiLRI (Simple Logic-based RDF Interpreter [18]). One of the most important differences to F-Logic and SiLRI is that TRIPLE does not have fixed semantics for object-oriented features like classes and inheritance. Its modular architecture allows such features to be easily defined for different object-oriented and other data models like UML, Topic Maps, or RDF Schema. Description logics extensions of RDF (Schema) like OIL, DAML+OIL, and OWL that cannot be fully handled by Horn logic are provided as modules that interact with a description logic classifier, e.g. FaCT [19], resulting in a hybrid rule language.

4.6.1 Namespaces and Resources

TRIPLE has special support for namespaces and resource identifiers. Namespaces are declared via clause-like constructs of the form *nsabbrev := namespace*, e.g.:

```
rdf := "http://www.w3.org/...rdf-syntax-ns#".
```


Resources are written as *nsabbrev:name*, where *nsabbrev* is a namespace abbreviation and *name* is the local name of the resource. Resource abbreviations can be introduced analogously to namespace abbreviations, e.g.:

$$\text{isa} := \text{rdfs:subClassOf}.$$

4.6.2 Statements and Molecules

Inspired by F-Logic object syntax, an RDF statement (triple) is written as:

$$\text{subject}[\text{predicate} \rightarrow \text{object}]$$

Several statements with the same subject can be abbreviated as “molecules”:

$$\text{Stefan}[\text{hasAge} \rightarrow 33; \text{isMarried} \rightarrow \text{yes}; \dots]$$

RDF statements (and molecules) can be nested, e.g.:

$$\text{Stefan}[\text{marriedTo} \rightarrow \text{birgit}[\text{hasAge} \rightarrow 32]]$$

4.6.3 Models

RDF models, i.e., sets of statements, are made explicit in TRIPLE (“first class citizens”). Statements, molecules, and also Horn atoms that are true in a specific model are written as

atom@model (similar to Flora-2 module syntax), where *atom* is a statement, molecule, or Horn atom and *model* is a model specification (i.e., a resource denoting a model), e.g.:

$$\text{Michael}[\text{hasAge} \rightarrow 36]@\text{factsAboutDFKI}$$

TRIPLE also allows Skolem functions as model specifications. Skolem functions can be used to transform one model (or several models) into a new one when used in rules (e.g., for ontology mapping/integration):

$$O[P \rightarrow Q]@\text{sf}(mI, X, Y) \leftarrow \dots$$

If all (or many) statements/molecules or Horn atoms in a formula are from one model, the following abbreviation can be used: *formula@model*. All statements/molecules and Horn atoms in *formula* without an explicit model specification are implicitly suffixed with *@model*.

4.6.4 Logical Formulae

TRIPLE uses the usual set of connectives and quantifiers for building formulae from statements/molecules and Horn atoms, i.e., \wedge , \vee , \neg , \forall , \exists , etc. All variables must be introduced via quantifiers, therefore marking them is not necessary (i.e., TRIPLE does not require variables to start with an uppercase letter as in Prolog).

4.6.5 Clauses and Blocks

A TRIPLE clause is either a fact or a rule. Rule heads may only contain conjunctions of molecules and Horn atoms and must not contain (explicitly or implicitly) any disjunctive or negated expressions. To assert that a set of clauses is true in a specific model, a model block is used: `@model { clauses }`, or, in case the model specification is parameterized:

$$\forall Mdl \text{ @model}(Mdl) \{ \text{clauses} \}$$

4.7 A Model for the Federation of Communities and Metaproviders

Many different Educational Nodes could join our Smart Spaces for Learning. Then it would be important to define groups and different levels of collaboration between them. Here there is a first draft of a generic model for community federation which allows to define communities in a distributed environment and allows communities in different servers, domains or geographic areas to cooperate and work together. The model is as follows.

The first element of the model is the *MetaProvider*. Each *Server*, *Peer* or whatever component which is host members which may offer educational resources to the communities they belong will be called a *MetaProvider*.

The functions of a *MetaProvider* are:

- To register users
- To manage *communities*

In the model each community is managed by a Community Manager, who is responsible for registration of users as community members. There may exist more than one manager. The community where a user is registered will be called his BaseCommunity.

Each community manages offers, bookings and deliveries among community members. Offers are made by community members. The scope and visibility of each offer can be defined by the provider. The scope may be groups of individual members, groups of communities or public (universal offerings visible to any member).

The purpose of this model is to define a federation model for communities or metaproviders which allows communities to integrate in groupings, with the purpose of allowing different levels of collaboration among the members of an the integrated community.

4.7.1 *Agreements among communities*

The community manager represents the community and acts as a delegate of the members of the community. Therefore federation of communities is based on *Agreements* made between the managers of two or more communities. Various types of agreements can be considered, for example

- ◆ *Inclusion agreement*: Integrates a community into another one.
- ◆ *Exchange agreement*: bidirectional exchange of goods.
- ◆ *Provision agreement*: unidirectional export, where only the offers of the providing community are visible in the receiving community.

Other types of agreements may exist.

The **inclusion agreement** is hierarchical and integrates a child community into a parent community. It allows the integration of one or more communities into a larger community. The new community will be formed by the union of the members of both communities. It is unclear if the parent community manager should get also a manager of the child community.

“**A includes B**” will be used as a shorthand for an inclusion agreement where A is the parent and B the child community.

This agreement creates a transitive inclusion relation (member set inclusion) among communities, where

A Includes B and B Includes C, implies A Includes C

The **peering or exchange agreement** is in practice very similar to the inclusion of communities, but is symmetric. This agreement will allow bidirectional exchange of community goods (offers). No new community is created and the managers of the communities will be the ones existing before the agreement.

“**A Peers B**” will be used as a shorthand for a peering agreement.

The peering agreement creates a commutative relation among communities, where:

A Peers B implies B Peers A

Transitivity of the peering relation is an open choice. If the peering agreement is defined as transitive large federations of offer exchanges can be created with minimal effort. Transitivity is defined as

A Peers B and B Peers C, implies A Peers C

If a non transitive peering relation is defined, each community controls the accessibility of offers, but creation of large federations will need a large number of agreements.

The **provision agreement** is hierarchical and relates a provider community with a provided community. The provided community will receive all the goods (offers) of the provider communities. No new community is created and the managers of the communities will be the ones existing before the agreement.

“**A ExportsTo B**” will be used as a shorthand for a provision agreement where A is the provider and B the provided community.

The provision agreement creates an export relation. The export relation is not commutative but a transitive relation is optional. A transitive export relation will facilitate the creation of large federation of exporting communities. This relation should be defined as follows

A ExportsTo B and B ExportsTo C, implies A ExportsTo C

Those agreements may be composed together. But composition rules should be clearly defined. For example: What happens if a community is imported which has peering of provider agreements with others? Or if a community which has provider agreements includes other communities.

4.7.2 Implementation

The agreements may have different types of implementation. An easy implementation of a federation of UBPs is outlined here, which is based on exporting metadata and allowing delegated access for booking and delivery.

The search queries should be distributed to the federation of communities or metaproviders, whereas the offers, bookings and deliveries are local from the implementation point of view. Remote members can access bookings and deliveries on remote servers with rights or credentials exported by BaseCommunity.

Delegated Access means that the base community of a user can provide access rights to the members of the base community for accessing federated communities in other servers.

Delegated rights or credentials will enable a member to access a community on a remote server to book an offer made in this community, as well as to participate in the delivery of a remote offer.

4.7.3 Example

M1 is a Metaprovider, for example a UBP at W.U.Wien, which creates the following communities

Community 1: CSWUW, the Computer Scientists Community at W.U.Wien

Community 2: CSJKUL, the Computer Scientists Comm. at Joh. Kepler Un. Linz

M2 is another Metaprovider, for example a UBP at Demokritos, which creates the following communities

Community 1: CSGreece, which groups all computer scientists in Greece

Community 2: CSNCSR, the Computer Scientists at Demokritos.

Community 3: CSNTUA, the Computer Scientists at National Tech. Univ. of Athens

Computer Scientists at CSNCSR, CSNTUA should also belong to the CSGreece community and should be integrated into it. Therefore two agreements should be made:

Agreement 1: CSGreece includes CSNCSR

Agreement 2: CSGreece includes CSNTUA

Example of Peering Agreements:

Peering Agreement 1: CSWUW peers with CSGreece

Peering Agreement 2: CSWUW peers with CSJKUL

User access examples: users U1 and U2 are registered in M1 (WUW)

U1 belongs to CSWUW community

U1 may access CSWUW as a member

U2 belongs to CSJKUL community

U2 may access CSJKUL as a member

Access allowed by Peering Agreements:

PA1: allows U1 to access CSGreece

PA1: allows U1 to access CSNTUA

PA2: allows U1 to access CSJKUL

PA2: allows U2 to access CSWUW

PA2: does not allow U2 to access CSGreece

4.8 Focusing Personalisation Services

In the deliverable D2.4 we have dealt with basic learner profiles based on learner modelling standards (see D2.4 or [13] for more discussion about standards and schema). We are working towards RDF bindings for the schema for learner profiles developed. The RDF features have interesting implications for learner modelling. They allow us to use schema elements of both standards and also elements of other schemas. The RDF models can be accessible by different peers and even more different peers can have own representation of opposite peer.

Personal learning assistants can be considered also as peers which represent users or a learner. The metadata about a user can be provided to other peers for computing purposes (when the user allows this sharing of profiles). These peers can send and receive messages when a common language is used. Queries can be forwarded to one or more peers.

Based on this, Figure 4.6 depicts our current ELENA architecture. Circles represent simple providers without reasoning capabilities. Rectangles represent peers, which are able to perform programs (personalisation services).

By personalisation service we recognize a functionality, which customises access to Learning Services and Learning Resources (in the context of the delivery of a Learning Service) based on learner profiles (Career Development Plans can be part of such a profile). The result of the personalisation service is usually a customised view on a Learning Repository or a Learning Management Network. The customisation can be performed in many ways using techniques such as collaborative filtering or rule-based filtering in order to modify a user's query or to reduce the results produced by the query.

Multiple rectangle symbols represent metadata. Learning resources are provided through resource provider peers. Resources can be referenced by courses which represent simple learning services. Courses can be personalized by adaptation services provided by personalization peers. Courses and resources can be recommended by recommendation services or can be filtered by filtering services. Personal learning assistants support learners or user to use the network.

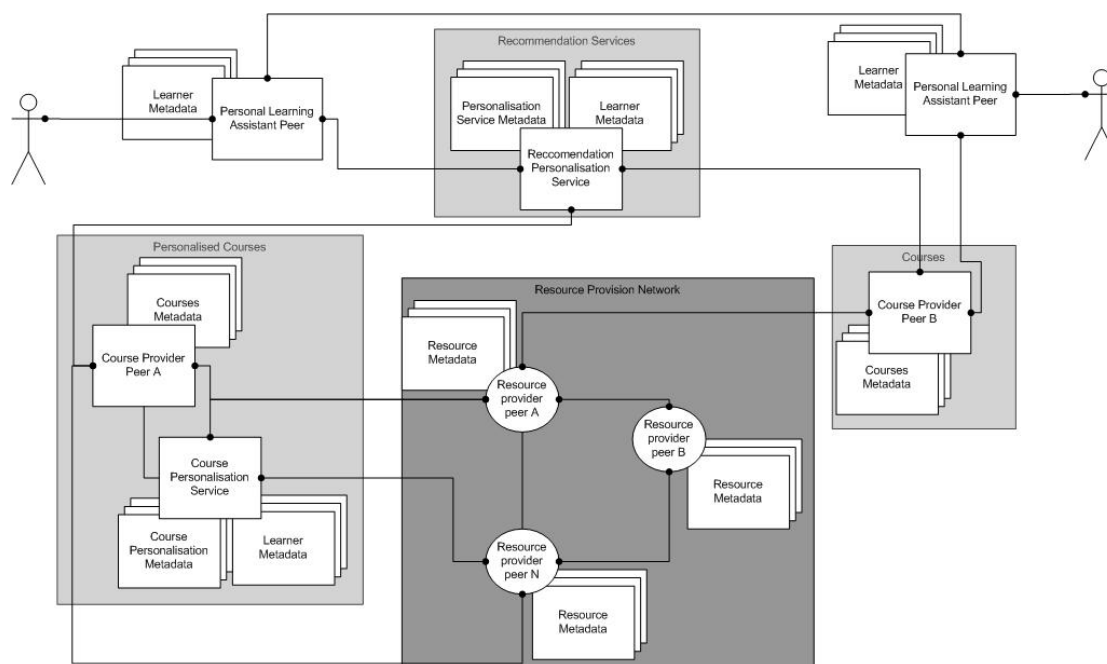


Figure 4.6: ELENA architecture for personalization services. The personalization services might be generic adaptive functionalities provided and described in common language, e.g. first order logic (see [14] for details). The generic personalization services can be then reused in several courses and/or queries. The example of such generic personalization service would be recommendation of particular course fragment based on its prerequisites what can be defined independently from topics and fragments available in the course.

The Datalog-based RDF-QEL language allows us to implement reasoning capabilities by using prolog or other Datalog-based systems. These reasoning can be run on specific peers when broadcasted messages are received or after receiving results of submitted query. In [15] we already described some reasoning capabilities for some adaptive hypermedia techniques for an electronic course. The basic idea is to match learner performance descriptions to requirements (accessibility restrictions) a resource has for specific learner. It means that both (learning resource and learner model) use one schema according to the requirements for adaptation based on learner performance within the domain. The performance schema was derived from PAPI standard.

Aggregated user profile of users who visited a resource or liked the resource can be modelled as accessibility restrictions as well. However, different features and structure of this constraint will be needed. This model can be matched to other users searching for resources according to their goals, interests, etc. This mechanism is also used in some collaborative filtering techniques and can be represented by reasoning programs at different peers.

5. Scenarios for Integration

There are three possible scenarios for integration:

- Embedding Edutella query service and provider interface into an educational node
- Invoking Edutella query service from an educational node
- Implementing a wrapper for accessing the metadata repository of an educational node

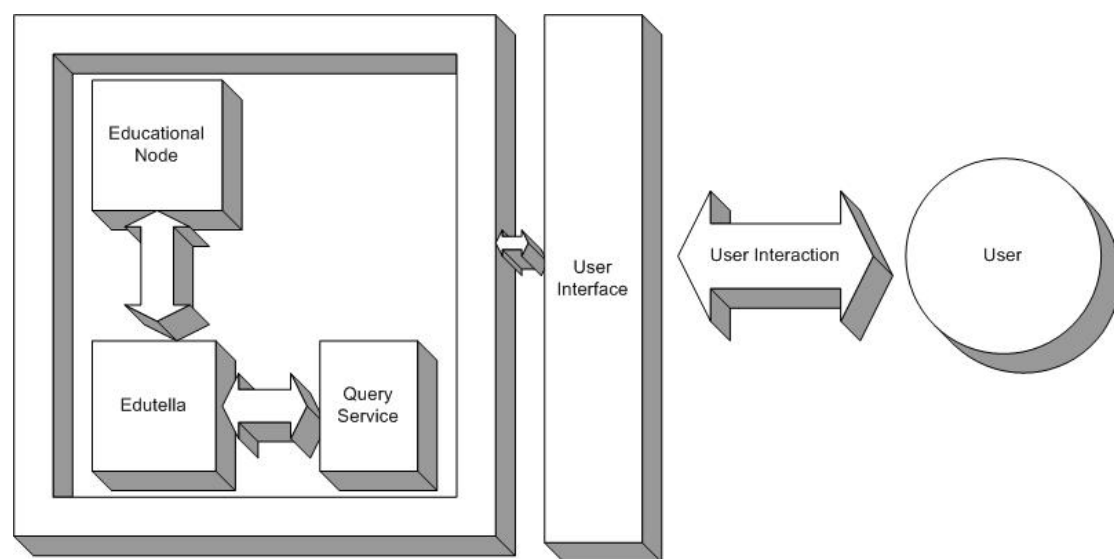


Figure 5.1 Embedding Edutella query service and provider interface into an educational node

A schema for the first integration scenario – Embedding Edutella query service and provider interface into an educational node – is depicted in fig. 5.1. This scenario assumes that the program code of Edutella will be added to the program code of the educational node. The Edutella services thus extend services provided by the educational node. This scenario can be taken into account for educational nodes, which:

- are programmed in JAVA
- provide JAVA interfaces to their repositories and
- enable to customise their source codes.

As we discussed in the previous section, file-based and relational database providers are already implemented and can be considered as the most natural choice for accessing metadata repositories of an educational node. If the educational node provides a different type of metadata repository from those mentioned, a

transformation procedure has to be implemented. The procedure transforms metadata from an educational node metadata repository schema to an appropriate schema for querying by Edutella.

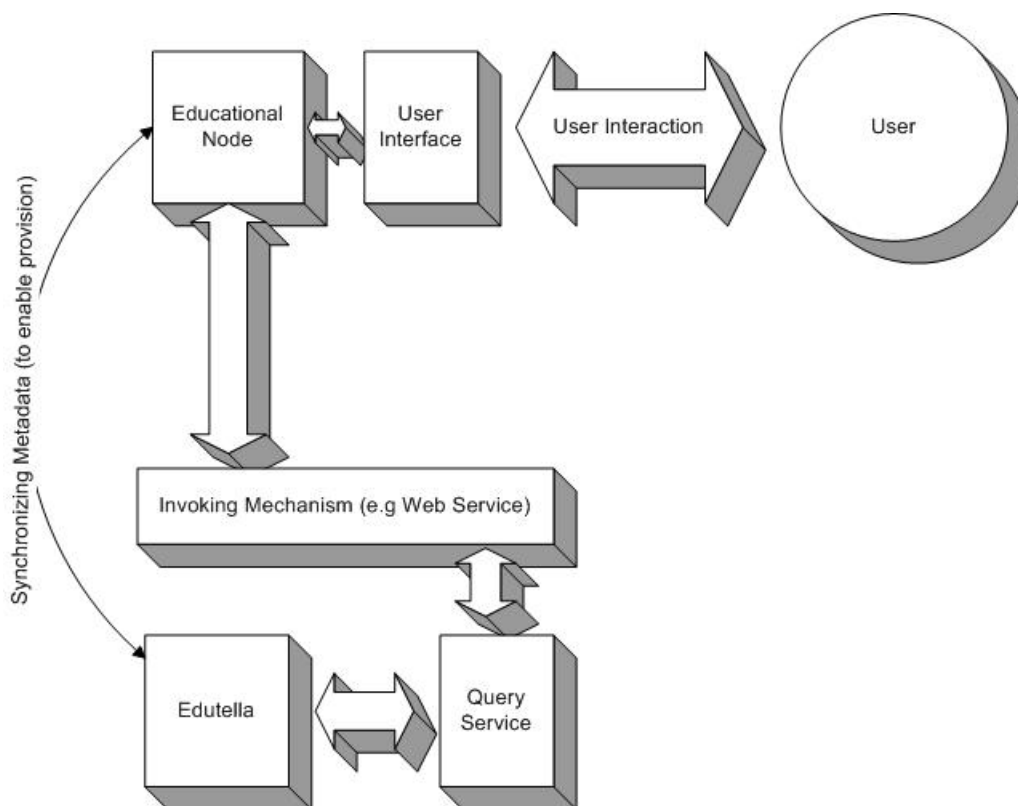


Figure 5.2 Providing and querying in the ELENA network from an external educational node

Figure 5.2 depicts a schema for the second scenario – invoking Edutella query service from an educational node. This scenario assumes a stand-alone Edutella peer and an educational node. The query service is provided through some kind of invoking mechanism (e.g. web service). If educational node metadata provision is needed, synchronization between the metadata repository of the educational node and the metadata repository of the Edutella peer must be provided. This scenario can be considered for educational nodes, which:

- Are programmed in a different language than JAVA
- Do not allow to customize its program code or
- Do not allow for extending the program code.

For the mentioned synchronization mechanism, an appropriate time interval for synchronization has to be assigned. If the educational node metadata repository schema is different from RDF, the transformation procedure has to be provided together with synchronization. The up-to-dateness of metadata at the Edutella peer should also be considered regarding the appropriate time interval for and between synchronization.

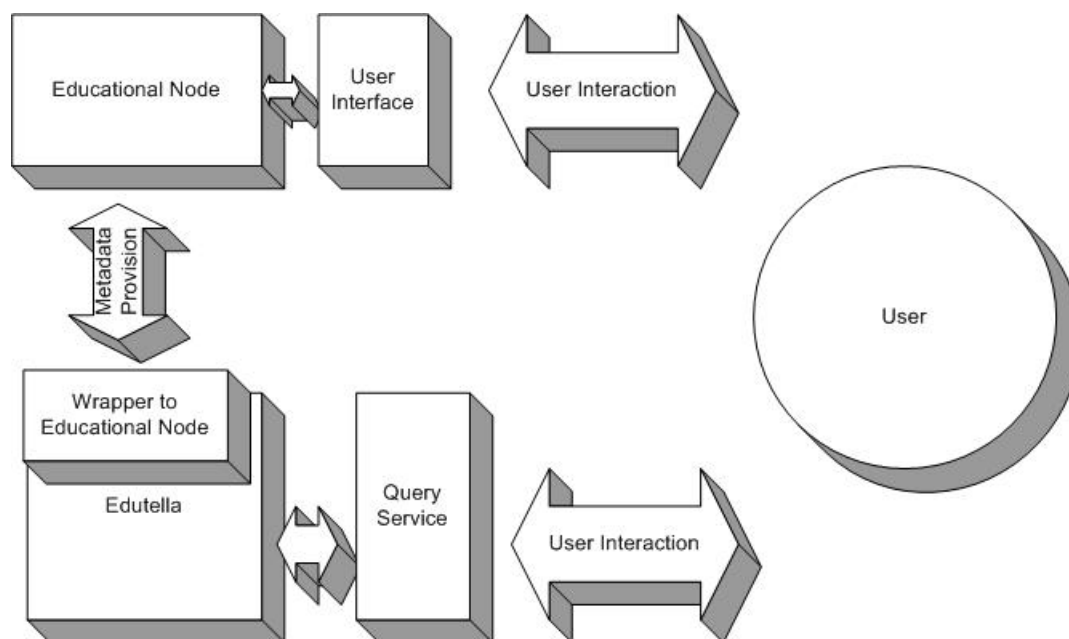


Figure 5.3 Wrapper access to the educational node

Figure 5.3 depicts a schema for the third integration scenario – implementing a wrapper for accessing the metadata repository of an educational node. This scenario assumes a stand-alone educational node and Edutella peer. The Edutella query service is provided to a user through an appropriate user interface. The metadata provision is realized through a wrapper to the metadata repository of the educational node. This scenario can be considered for educational nodes, which:

- Provide their metadata repository schema specification
- Provide metadata repository access programming interface appropriate for the JAVA programming language, or
- Do not allow to customize the educational node program code.

If the educational node metadata repository schema is different from RDF, a transformation procedure has to be part of the wrapper.

6. Interfaces

Figure 5.1 depicts a schema for one Educational Node from the software component point of view.

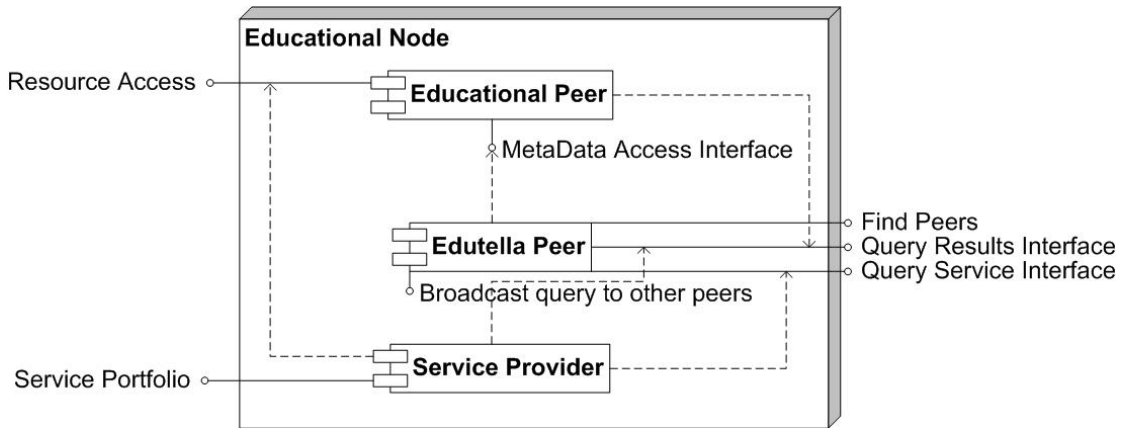


Figure 6.1 High level component architecture of educational peer in smart learning space

There are three high level components depicted in fig. 6.1:

- Educational Peer – any system from the ELENA consortium or possibly future system, which provides content and metadata,
- Edutella Peer Provider – standard Edutella provider, and
- Service Provider – optional (not implemented yet) component, which provides integrated services of Edutella peer and content management system.

Educational Peer will provide interface to its content and metadata. The Metadata Access interface is used by Edutella Peer component. This interface comprises the metadata query interface and the metadata results interface. The SQL or file based access to metadata is preferred. If there is another interface, the wrapper should be implemented as an implementation of ProviderConnection interface (see Appendix B and C). The Edutella Peer provides Find Peers interface, Query Results interface and Query Service interface. They can also be used outside of the Educational Node. The Edutella Peer can stand as a metadata provider (can be queried) or consumer (consumes query results from broadcasted query to other peers – Broadcast query to other peers interface). The Service Provider component can be developed, which will provide access to a service portfolio. This interface can also be used outside of the educational peer. The Service Provider component accesses the Query Service and Query Results interfaces to be able to compose the service portfolio. It also uses the Resource Access interface to be able to provide resources within the educational service.

The integration scenarios discussed in previous section are possible realizations of usage relationships (dashed arrows). The implementations of simple `ProviderConnection` interface and `QueryService` interface is discussed in Appendix B. The specifications of that interfaces are in Appendix C.

7. Network Nodes and Artefacts

The A&S network currently consists of several different nodes which are connected via the Edutella peer-to-peer (P2P) network. A special educational node in the network is based on the UNIVERSAL brokerage platform (UBP). This broker instance offers an interface for automatic provision of learning services via web services. Currently three different Learning Management Systems (LMS) implement these web services to automatically provide their resources to one of the brokers:

- IMC Clix
- Arel
- CDI ITeachYou

Another node in the network is an instance of an Edutella file-based provider.

At <http://nm.wu-wien.ac.at/e-learning/interoperability/> a Java client can be downloaded, which allows you to test the UBP interface for providing learning resources.

By utilizing these web service interfaces resources can be announced to a wider public within the Edutella-based P2P network. All Educational Nodes connected to the Edutella network can be queried by downloading an Edutella consumer (e.g. the DC Consumer) available at <http://edutella.jxta.org/downloads/index.html>. An instance of a Personal Learning Assistant (PLA) allows learners in searching for, selecting and contracting Learning Services.

7.1 Educational nodes and artefacts

This section provides high-level description on systems as educational node artefacts in Elena network. This description comprises:

- Basic information about the educational node,
- Metadata artefacts,
- Resource and educational service artefacts, which are currently available,
- Mapping rules, for transforming non LOM based schemas to LOM, and
- Current state of implementation.

7.1.1 EducaNext

Educanext is a web-based platform which supports the creation and sharing of knowledge. The portal is based on the Universal Brokerage Platform (UBP), which enables collaboration among educators by providing a full range of services to

support the exchange of Learning Resources. The portal can be accessed at: <http://www.educanext.org/>.

Educators are able to provide learning resources to the UBP and specify offer conditions on which consumers, who are also educators, are required to agree before accessing the learning resources. Based on educational metadata and target-audience specific offer information (e.g. commercial offer, open source-like licensee agreement), learning resources are advertised through the UBP catalogue and interest-specific mailing lists. Based on this information, educators can choose and access learning resources from dispersed delivery systems such as video conferencing applications, learning management systems, streaming media servers and standard web servers after agreeing on the terms specified.

The UBP as a metadata management system primarily handles metadata descriptions of learning resources. The metadata are synchronised with the systems connected with the UBP. These systems manage data (information, content, etc.) and they provide metadata descriptions for the broker. The broker provides an interface for provision, access control, access and delivery, learning resource management, and inspection application services.

Artefacts Descriptions at Educanext

The UBP provides functionality for cataloguing and delivering both educational material and educational activities. Hence, two different kinds of learning resource descriptions are handled by the UBP. Educational activities are defined as instructor-supported educational events taking place at a specific (virtual) place and time. Educational material is defined as any kind of artefacts supporting educational activities (see Appendix D for more examples of artefact descriptions provided in RDF).

The broker can handle LOM and Qualified Dublin Core based XML descriptions. Appendix A provides you with sample records of learning resource descriptions for both, educational services and educational material.

The LOM field learning resource type is used to indicate whether a learning resource is an educational activity or an educational material. Depending on the learning resource type different values for media type, the learning resource type and contributor role are supported (see Table 1). For the discipline values the Dutch Basis Classification (see http://www.kb.nl/kb/resources/frameset_kb.html?/kb/vak/basis/bc98-en.html) is used.

	Educational Material	Educational Activity
Media Type	MIME Type	Application Sharing, Audio Conference, Chat, Classroom, E-mail, Mailing List, News Group, Video Conference, Unknown, Other

	Educational Material	Educational Activity
Learning Resource Type	Case Study, Case Study Guide, Collection, Data Set, Demonstration, Educator's Guide, Exam, Exercise, Experiment, Figure, Lecture Notes, Narrative Text, Presentation, Problem Statement, Questionnaire, Recorded Lecture, Reference Material, Research Paper, Research Study, Self Assessment, Simulation, Text Book, Thesis, Tutorial, Unknown, Other	Case Study, Course, Course Unit, Exam, Exercise, Experiment, Group Work, Lecture, Presentation, Project, Unknown, Other
Contributor Role	Author, Content Provider, Editor, Instructional Designer, Technical Implementer, Coordinator, Course Administrator, Instructor, Lecturer, Teaching Assistant, Tutor, Unknown, Other	Content Provider, Instructional Designer, Technical Implementer, Coordinator, Course Administrator, Instructor, Lecturer, Teaching Assistant, Tutor, Unknown, Other

Table 1: Attribute Values for Media Type, Learning Resource Type, and Contributor Role

Artefacts Provided

About 111 learning resources are currently listed in the Computer Science discipline and can be provided as artefacts to the smart learning space (accessed at 26/11/2002), but only for some the filtering rules (see next subsection) apply. A list of learning resources in the Computer Science domain handled by Educanext can be accessed at: [http://www.educanext.org/UNIVERSAL/servlet/LRSearch?pageID=srchBrowseCatalog&\\$fCatID=54](http://www.educanext.org/UNIVERSAL/servlet/LRSearch?pageID=srchBrowseCatalog&$fCatID=54)

Filtering and Mappings

Only learning resources having the Computer Science classification assigned are relevant for the smart learning space. Those learning resources, which do not comply with the definition of educational service, have to be filtered out as well. In addition any kind of educational material, which is not designed as self-study material, has to be filtered out.

As a consequence the following educational material types should not be provided to the smart learning space: Case Study, Case Study Guide, Collection, Data Set, Demonstration, Educator's Guide, Exam, Exercise, Experiment, Figure, Lecture Notes, Narrative Text, Presentation, Problem Statement, Reference Material, Simulation, Thesis, Questionnaire.

The following types of educational material comply with the definition of an educational service, in general: Recorded Lecture, Research Paper, Research Study, Self Assessment, Text Book, Tutorial,

The following types of educational material require manual examination: Unknown, Other.

Implementation

The EducaNext platform as one of the two UBP instances in the network is available under <http://www.educanext.org>. All the educational material and educational activities available on EducaNext are also available via the peer-to-peer network. Thus all artefacts stored in EducaNext are currently provided *unfiltered* to the Edutella network.

The EducaNext portal implements the synchronization interface via WSDL and SOAP. WSDL web-service description files of the synchronization interface can be downloaded at <http://www.educanext.org:8080/services/synchronization?wsdl>.

The server that hosts the EducaNext portal also runs an Edutella OLR provider that is connected with a special triple-table of EducaNext in order to enable Edutella consumers to query EducaNext. The name of the provider in the P2P network is "EducaNext".

7.1.1.2 Clix

CLIX is a Learning Management System developed by IMC. CLIX provides education services for learning service providers. It stands for Corporate Learning and Information Exchange. CLIX is a software application with which you control all information, learning and knowledge processes via your browser – in the intranet or internet, in real-time without add-ons and plug-ins.

Artefacts Descriptions in Clix

CLIX offers metadata compatible with DublinCore and LOM standards. You can add metadata tags according to your needs. The Metatag Manager as a CLIX component allows:

- The flexible definition of metatags
- The individual extension of existing metatag sets and
- The free keyword assignment to all content-related objects in CLIX.

The metatags that CLIX bring as default are:

- Administrator – Administrator for CLIX content, CLIX users etc.
- Cancellation by participant – Cancellation by participant
- Comment – Comments on an object
- Cost – Cost
- Course Creator – Course Creator (AICC Parameter)
- Course ID – Course ID (AICC Parameter)
- Course System – Course System (AICC Parameter)
- Creator – Creator of object
- Credit points – Credit points for participants in instruction such as courses, lectures etc.
- Deadline – Deadline
- Description – Description of an object
- Description – Description
- Developer ID – Developer ID (AICC Parameter)
- Display File – File to display the exercise
- Duration – Duration
- Educational objective – Educational objective of course
- End Date – End of a course, of a chat etc.
- Image – Graphics file in a participant's profile, media element etc.
- Keywords – Keywords for searching for objects
- Keywords – Keywords
- Level – Course Level (AICC Parameter)
- Max Fields CST – Max Fields CST (AICC Parameter)
- Max Normal – Max Normal (AICC Parameter)
- Moderator – Moderator e.g. of a chat
- Release Date – Release Date of the worksheet

- Requirements – Requirements
- Return Date – Return Date: End of the correction period
- Source (URL) – URL of a file from the Internet or a network
- Start Date – Start date of a course, chat etc.
- System ID – System ID (AICC Parameter)
- Title – Name of an object, e.g. a media element
- Title – Title of the component
- Total AUs – Total AUs (AICC Parameter)
- Total Blocks – Total Blocks (AICC Parameter)
- Tutor – Tutor for a course or course module
- Type of order – How course was booked, such as own booking, third party booking etc.
- Upload File – Upload file in order to add an object
- Version – Version (AICC Parameter)
- Welcome text – Welcome text for course participants
- Max. participants – Maximum number of course participants

Artefacts Provided

IMC provides 8 main learning units in Business Engineering Basics subject, 11 main learning units in Business Engineer's methods subject, and 11 main units in Business Engineer's applications subject. Each unit is described by title, goal, main description, main content, and time required for this unit. This subject forms the "Business Engineer IHK" study program provided by IMC (see Appendix F for more information).

Filtering and Mappings

Because CLIX does not provide metadata conforming to RDF based LOM or the DC schema, a mapping between LOM and CLIX metadata schemas will be necessary. Only computer science learning resources are relevant for the first prototype. The filtering should be applied for other types of resources. The topic ontology used in CLIX should be mapped to ACM ontology.

Implementation

The Synchronization interface has been integrated into the Clix LMS. Herewith it is easily possible for providers working on Clix to publish their learning material on a UBP instance via the SOAP interface.

After logging into the system, please select "Content management → Courses → Course manager". Here at the top button row, one can see the symbol "Sync UBP" which becomes active as soon as a course is selected. By pressing the "Search" button, a course should become visible (see Figure 7.1 below). This course can be then synchronized with the UBP, by selecting the course and pressing the "Sync UBP" button.

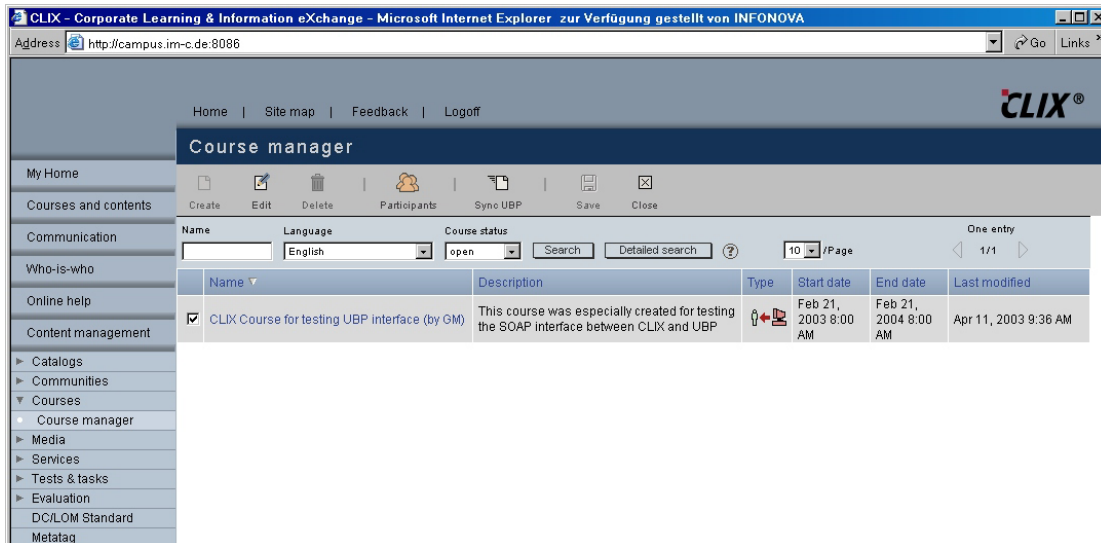


Figure 7.1: The course manager view in IMC Clix

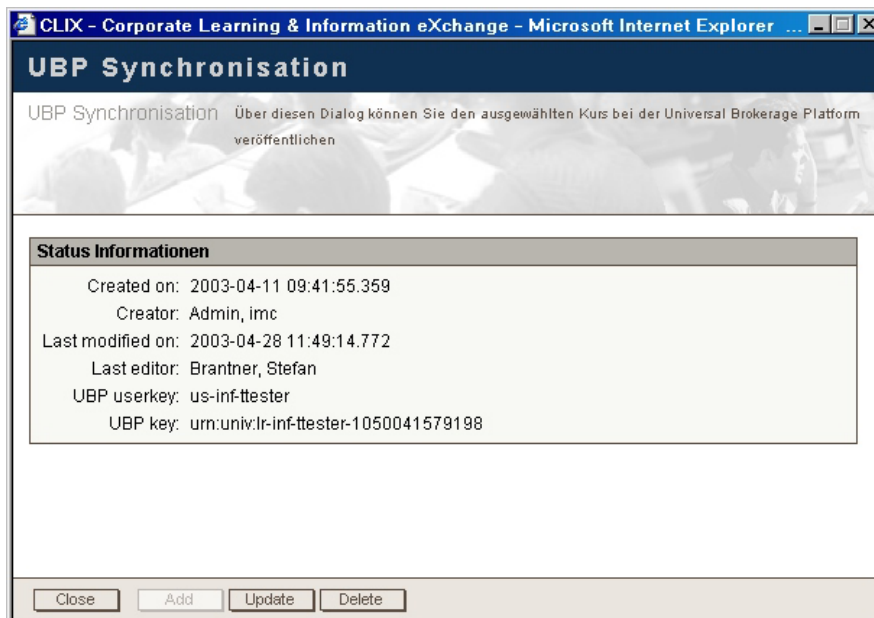


Figure 7.2: The dialog for synchronizing a course with the UBP

The next window (see Figure 7.2 above) offers to add, update or delete the course at the UBP. This instance of Clix is connected with the experimental UBP mentioned

above. If you remove the test course via the "Sync UBP" dialog, then the resource should be deleted from the experimental UBP, etc.

7.1.1.3 AREL – IDEAL (Interactive Distance Education and Learning) suite

Arel's IDEAL offers a unique training solution for corporations, distance learning institutions and large organizations. The IDEAL system enables top experts to deliver live and on-demand interactive broadcast sessions from a centre to a large number of participants in AREL Virtual Class sites and AREL Spotlight desktops via Intranets / Internet.

IDEAL is equipped with a build-in database to handle the administration and control of vast amounts of content. The Learning Manager™ includes the Arel Author™ for building course content, Arel Evaluator™ for assessing performance and Arel Reporter™ for generating reports.

Arel's innovative e-learning suite consists of three browser based applications: Arel Campus™, to login courses and retrieve training information, Arel Spotlight™ to actively participate in live interactive sessions and Arel Spotlight on Demand™ to take recorded interactive courses at your own pace.

Arel Campus extends the reach of corporate and educational infrastructures, by providing users with access to both personal and administrative learning data. Using the Campus, students can launch on-demand courses, attend a live e-learning session, preview course content, take off-line on-demand quizzes and communicate with instructors and expert assistants via text messaging and chat. Arel Campus provides data on course details and scheduling, registered participants, instructors, individual progress, grades and accreditation.

Arel Spotlight is a browser-based application that combines quality video, two-way audio and data to enable synchronous training at any desktop connected to the Internet or Enterprise Intranet. In addition to presentations, multiple-choice questions and tests, Arel Spotlight offers live collaboration tools such as application sharing, discussion groups, chats and guided browsing.

Arel Spotlight On-Demand extends the benefits of live learning, enabling asynchronous transmission of self paced and highly interactive sessions. Full recording of live sessions ensure the delivery of rich video content, capturing all the highlights and original interaction from the live session. Participant enrolment, registration, testing and grading are stored in the database for maximum traceability.

Artefacts descriptions at AREL IDEAL Suite

Course administrators, system administrators, and instructors use the IDEAL system for comprehensive registration, authoring, analysis and generation of reports. More details on the Administrative part are the following:

The 'Admin' menu has the following options:

- Users: Allows the creation of new database users
- Personnel: Provides an entry screen for studio and remote site personnel
- Studios: Allows registration of different IDEAL studios
- IDEAL Components: Allows pre-registration of IDEAL equipment including the Lesson Control Unit(s), Comm Server(s), SME Server(s), and Phone Process Lines
- System Tables: Allows updating voice languages and determining the location to save the participant's photos
- Company: Provides a linked company entry for scroll-down detail screens
- Job Title: Provides a linked job title entry for scroll-down detail screens

Courses, participants and other artefacts are registered, using the application and the information is updated on a central database server. The course content including session script and interactive questions is stored in the database also. During the interactive sessions, the participants' responses to multiple-choice questions are updated in the database. The responses can be viewed for real-time analysis by the instructor or used to evaluate specific data after the session. The Learning resources and the educational activities that can be handled in the Ideal System are the following:

Registration:

Entering data in the Ideal System is performed in two different modes, pre-registration and registration. Pre-registration of IDEAL equipment, studios, personnel and instructors requires filling in essential details. Groups can be created to facilitate simple registration of participants in the same category to courses. Administrators create the group names according to their specific needs. They can be names of departments, regions, time schedules, projects, or any other criteria. Course registration including lessons, tests, references, scheduling and other details can be stored in the database. Participants are registered with their personal details including course selection. Remote sites are registered with specific information.

Authoring:

The instructor's notes and/or session script are important information that can also be stored and can contain attached files. Session script provides an order of events for interactive session. Tests are prepared and are assigned an evaluation weight. Each Question can have a specific weight.

Courses:

The courses consist of lessons and tests

Course Level	Description
--------------	-------------

Course_id	Course Id
course_system_id	
title	Title
credit	Credit assigned to the course
description	Description of the course
lecturer_system_id	
studio_id	Studio's ID (assigned to the course)
course_type	What kind of course it is
attachment_system_id	Attachment of file for the course
helpdesk	Helpdesk
language_id	Specify language
vox_language_id	
group_id	Specifies specific group of students
internet_access	If students have internet access
record_status	Record
extra_field	Additional information
perform_status	Performance status
num_registered_students	Number of registered students
course_grades_type	Course grade types
performance_weight	Performance weight
passing_grade	Passing grade assigned to course
max_participants_in_lesson	Max nrs of participants in a lesson
sme_email	Subject Matter Expert email
published	Published course for viewing
published_date	Published dates for viewing a course

Lessons:

A course can have one or more lessons. Each Lesson is scheduled at a different time.

Course Lesson	Description
course_system_id	To which course it belongs
lesson_num	Lesson number
title	Lesson title
description	Lesson description
attachment_system_id	If file is attached to the lesson
studio_id	Which studio is assigned
language_id	On what language the lesson is
vox_language_id	
helpdesk	Helpdesk
record_status	Record status
extra_field	Extra fields
perform_status	Performance status
notes_edited	Notes edited
questions_edited	Questions edited
lesson_curving_range	Lesson curving range
lesson_type	Lesson type
lesson_flow_updated	Lesson flow- if updated
lesson_unique_identifier	Lesson Unique identifier
default_view_mode	Default view of lessons
lesson_flow_edited	Lesson flow edited
If_root_attachment_system_id	If root has an attachment
allow_view_content	Allow the viewing of lesson content
published	Published lesson for viewing
lesson_xml	Lesson xml file

Other General Lesson Attributes	Description
lesson_num	Lesson number
lesson_flow_title	Lesson flow title
tree_node_type	Lesson node
video_size	Size of the video
duration_in_minutes	The duration in minutes
weight	Lesson weight
completion_status	Completion status
notes	Notes

Tests:

Typically, a course has one or more tests. However, a course does not necessarily have a test assigned. Tests are scheduled during a lesson and can also be taken during the On Demand fashion.

Tests	Description
course_system_id	Course system
quiz_id	Test id
curving_value	Curving value
curving_type	Curving type

Tracks:

Organizations constantly need to train large audiences on a specific issue. The audience may be available at different times, based on shifts, time zones or availability of facilities. Different departments, divisions or instructors may also want to separate their course identification from other courses. The track feature allows an organization with such a need, to offer uniform training to their remote participants. Registering a “track” allows the same course content to be rescheduled.

Tracks for students	Description
course_system_id	To which course the track belongs

track_num	Track number
student_sys_id	Student system id
performed_quizzes	Performed quizzes
questions_answered	Questions answered
meetings_attended	Meeting attended
avg_quizzes_grade	Average quizz grade
avg_questions_grade	Average question grade
total_hand_raise	Total hands raised
total_connect	Total connections
extra_field	Extra field
avg_total_grade	Average total grade
record_status	Record status
remarks	Any remarks
avg_quizzes_grade_na	Average quizz grades
avg_questions_grade_na	Average question grade
avg_total_grade_na	Average total grade

Artefacts provided

Arel Ideal Suite is currently installed to the National Technical University of Athens (NTUA) - Laboratory of Biomedical Engineering and it is used for training the doctors of different hospitals in the use of Information technologies and in the operation of complicated examination machinery. Till now (27-11-02) there are 12 complete courses listed that combine different thematic areas on the specific subject from “Introduction to the Windows XP operating system” to “the Step by step Guidance to the use of Radi-Tomographer”.

Filtering and Mappings

Only learning resources having the Computer Science classification assigned are relevant for the smart learning space. Those learning resources, which do not comply with the definition of educational service, have to be filtered out first. In addition any kind of educational material, which cannot be delivered as self-paced course will be excluded. As a consequence only the recorded lessons provided from the Arel Spotlight On Demand, which are contained from a full package of video, audio, and

HTML synchronised files will be compliant to the definition of the educational service of the smart learning space,

Implementation

The integration with the UBP has been implemented successfully and the learners will be able to view the AREL content through the UBP.

7.1.1.4 IteachYou

IteachYou [6] is an independent platform and multimedia application, which is designed for use in the internet or intranets, and it does not use plug-ins to a great extent. It can be considered as a presentation template for the voluminous library of highly structured CDI content foreseen for IT, ERP and key qualifications.

At the same time IteachYou is a learning management system offering the following facilities: Basic unit of IteachYou are *lessons* = learning units of ca. one hour duration referring to certain content, described by a set of metadata (see below) in an unequivocal way. Each lesson consists of the following components:

- Topic
- Procedure
- Exercise
- Test

Administrators or teachers (who are modelled as particular roles in the system) can construct any kind of courses by combining lessons. Courses are also defined by metadata. A course is a sequence of lessons saved in a database as a chain of lesson IDs.

User roles are defined as follows: Administrators can install *users* in the framework of the system; but it is also possible that users are installed automatically (e.g.) by HR systems. Aiming at the same learning objective, as a rule, users are *grouped*.

Due to the character of the target group to which IteachYou is addressed, it is not possible that users register themselves at the moment. In the first place, the system is designed for the following purposes:

- for the application in the framework of CDI's 21 branches: According to the current training demand, the system shall help to carry out training as taylorised and modularised as possible, thus making a highly efficient use of time and training content,
- for the application in the framework of SMEs as a thin client solution to enable human resources development, possibly linked to other products.

CDI did not consider it as necessary to integrate a billing functionality under these conditions. Nevertheless, a service of this kind could be added without problems, allowing users to register themselves under the condition that an e-commerce component is integrated and an according business model is available. Metadata information enabling account referring to user and learning unit has already been implemented in the system.

Artefacts descriptions at IteachYou

As described above, the main functionality of IteachYou is to deliver highly taylorised training content automatically to the user, thus enhancing efficiency of training and minimising time spent for this purpose, thereby allows reducing costs.

Pre-selection of content, as a rule done by administrator or teacher, is carried out on the basis of a 3-stage descriptors concept: Every learning unit is described by the following metadata:

- **DESCRIPTOR** – describes the content focus of a lesson,
- **FIELD** – stands for the special field, which the learning unit is assigned to. For IT CDI has set up a catalogue of 15 terms (as database applications, networking, etc.),
- **QUALIFICATION** – stands for the training objective of the learning unit. This can be a (producer) certificate (as MCSE, ECDL), or client specific training goals.

Selecting content, trainers and administrators may use these criteria, eligible in a mask, as they like it. In the framework of a sector, selected terms are connected by an OR-connector, by an AND-connector if sector limits are crossed. In order to adapt content as far as possible to the individual user needs, the ITeachYou-Server uses a *skill-gap-analysis*: ca. 10 different questions are assigned to every learning unit to cover all didactic needs, the following question types are implemented:

- multiple-choice MC with simple or multiple choice
- blank-question Fill in free text (words)
- fill-in-question Fill gaps with words
- completion-question Fill gaps via drag and drop from a list
- reorder-question Sort list via drag and drop
- assignment-question Assign terms via drag and drop
- claim-question True/false statements
- pull-down-choices Select solution from pull-down list
- drag-drop-question Drag and Drop with fields and/or tables

If a user starts the system by using a password and his log-in name, he automatically gets a list of learning units selected by the trainer/administrator. The user may now decide if he wants to work through or if he wants to pass the tests which refer to this content, thus enabling him to define his knowledge level. In the latter case a new personalised course will be generated after the evaluation of the test, consisting exactly of the learning units for which the user's knowledge could not be recognised as sufficient. All content provided for the user can now be used at the screen and converted into printable PDF-files.

Described tests can be carried out several times, and it might be assumed that the extent of new generated courses will be reduced from one time to the other, so that the user will be able to test his level of knowledge continually. There will be provided suggestion for the answers as well, but if one uses this information resource this will reduce the number of *credit points* achievable by passing tests. (If a solution is demanded *before* answering a question, the number of achieved credit points, of course, is 0.)

Test may also be carried out in the *exam simulation mode*. This mode allows only one test per user, to be carried within a (variable) time limit.

There might also be carried-out tests which are fully separated from content (e.g. aptitude tests, language tests). For that purpose an interface is provided where trainers may include their own tests (+ necessary metadata) Administrators and trainers may use this pool of test questions and select some of them to set up new exams. These exams will be described by metadata delivering information about, which will give information about:

- target audience
- time and place of production
- training institution (certification - target) of participants, etc.

All contents are described by a set of metadata compatible with LOM and IEEE. The user cannot add some metadata, as IteachYou, as a rule, is delivered with content of CDI. This content has not only to be evaluated by the server, but is also relevant for CDI's -own content management system.

The ITeachYou Server uses the following metadata, oriented to LOM and IEEE LTSC 1484 and adapted to CDI's needs:

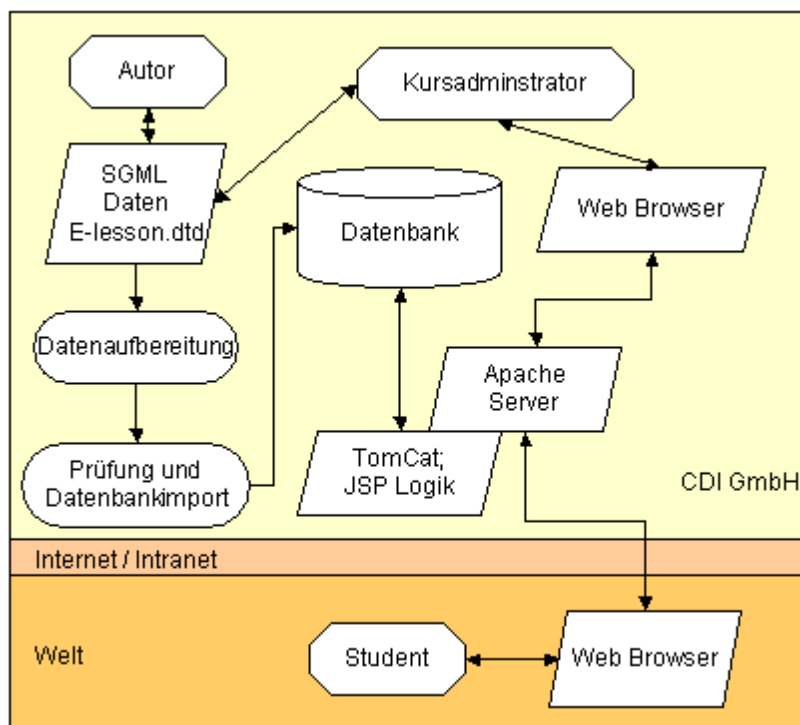
Name	Description
On lesson level:	
Title	Title of lesson
Subtitle	Subtitle of lesson

Author	Author	
Owner	Owner of intellectual property rights	
Document-Type	Document type; CCIT, CGM, CGM-BIN, CGM-CHAR, CGM-CLEAR, DXF, EPS, GIF, HTML, JPRG, PCX, SGML, XML, TIFF,	
Document-Identifier	Automatically generated document identifier	
Pre-Version	Pre-Version, reference to DB-ID	
Cost-Base	Price or calculation factor for invoice	
Abstract	Short content abstract	
Version	Current version	
Generated on	Date of current version	
Valid by	End date of current version	
Status	Document status: Development/ piloting / release	
Links to	Possible links to external documents	
Defined Ids	<i>Not used</i>	
Prerequisites	Necessary pre-knowledge for this learning units; reference to lesson-ID(s)	
Objectives	Objectives of learning unit	
Classes		
	Descriptors	Description of content
	Qualifications	Training objectives / Qualifications / certifications for which this learning unit is needed
	Fields	Knowledge fields to which the learning unit is assigned (CDI has defined 15 categories from the IT area)
External Refs	Possible references to external (client-owned) documents	
User-Data		
User-ID		
Userlogon		

UserPWD	
UserFName	
UserMName	
UserLName	
UserEMail	
UserIsConnected	
UserStreet	
UserState	
UserZip	
UserCity	
UserGender	
UserIsAdmin	Assigns administrator rights to user
Course data	
CourseID	
CourseTitle	
CourseLevel	Low, middle, high
CourseDesc	Course description
CourseVerion	
CourseSystem	At this moment, only CDI
CourseMaxNormal	
CourseDateCreated	
CourseRequirement	
Data in the sector of tests	
CurrLesson	
ForCredit	
HadTest	Test has already been carried out?

TestOnce	Test must be carried out only one time (in the exam simulation mode)
TestFinish	Test was interrupted / not interrupted
Teststart	Start time of test
Testend	End time of test
TestAreaID 1 + 2	Reference to specific field of test (for selection purposes)
Aspect	For client-specific customizing; currently not used
Attempts	Admitted number of trials
Author	
Credit	
Id	
Score	Low, middle, high
Version	
Data in the sector of exam simulation	
ExamID	
ExamDate	
ExamPeriod	Period of validity of exam
ExamScore	Necessary number of points
ExamFile	Export file for test
ExamTestCounter	
ExamTitle	

The application has the following architecture (simplified graph):



The system has the following attributes (summary):

- Text basis in SGML or XML
- Semi-automatic conversion of previous data and method to integrate previous data into the new system
- Definition of courses
- Definition of questions (tests) and exams
- Implementation and integration with a database
- Administration of students, courses and user groups/user roles
- Server-driven sequence of lessons
- Server-driven tests and evaluation of tests
- User profiling

Artefacts provided

Currently existing content-basis comprises ca. 180 lessons, mainly from the IT and key qualification area. A part of this content is connected with flash-animations and video sequences (for which plug-ins are required).

The user of IteachYou disposes of the following communication facilities:

1. *User home-page*, presenting the following data after the user has logged in
 - history-List
 - date of last access
 - courses for which the user is registered
 - news for the user
2. *Forum* where administrators, trainers and users can include and exchange information referring to training; an animated “learning companion” informs about new contributions
3. a *messenger* which can be used to set up contacts to trainers and other members of the work group, at the same telling who is currently online.
4. a *chat-forum*
5. a *Video conference system*

Filtering and Mappings

Only learning resources having the computer science classification assigned are relevant for the smart learning space. Because IteachYou uses LOM based metadata schema but not in RDF, the mappings to RDF are needed. Because IteachYou uses finer grained topic classification than for example the assumed ACM classification, mappings between IteachYou topics and the ACM ontology topics will be provided. The resources, which do not belong to the topic from that ACM ontology will be filtered.

Implementation

In ITeachYou the Synchronisation interface has been implemented as a menu point of the standard administration front-end on the ITeachYou test-server. The test-server is accessible at <http://cmi.cdi.de:8080> . After logging into the system with an administrator account, please select “ELENA-Projekt” (see Figure 7.3 below).

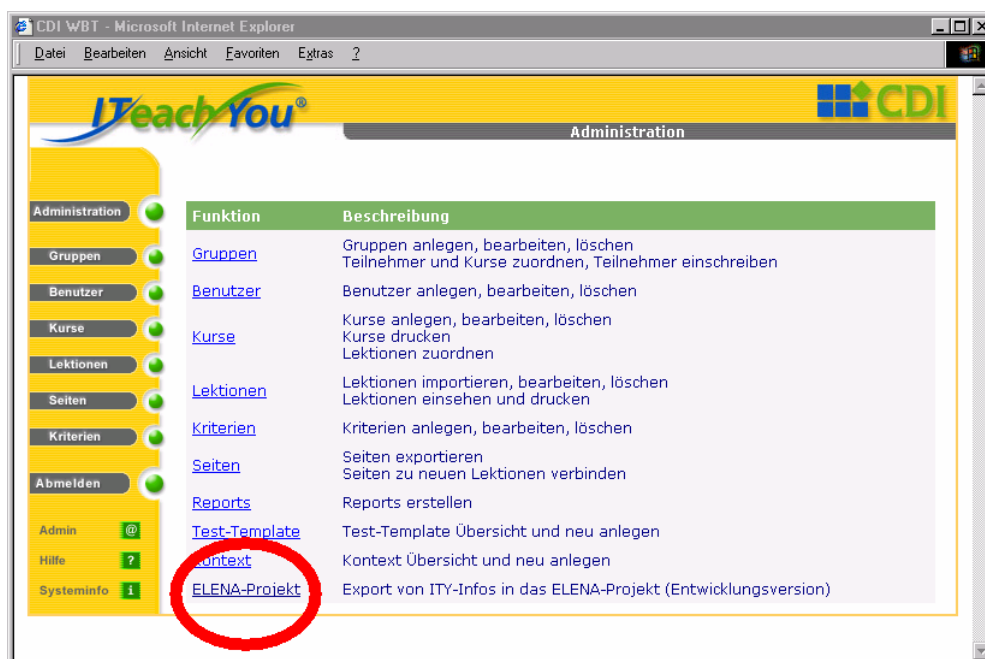


Figure 7.3: Administration menu

After having chosen “ELENA-Projekt” you will see the main menu of “RDF-Export” (see Figure 7.4). Here you can decide if you want to create an XML-File of a course or an exam, if you want to synchronise already created XML-files with the UBP-server, or if you want to download created XML-files to place them wherever you want.



Figure 7.4: Main menu of “RDF-Export”

If you choose “Kurs” you can select the course, in order to create the XML-file in the next window (see Figure 7.5).



Figure 7.5: List of courses which can be used

After having chosen a course (or exam) a window (Figure 7.6) will be opened automatically where you can choose one ore more Codes of the Dutch Basic Classification.

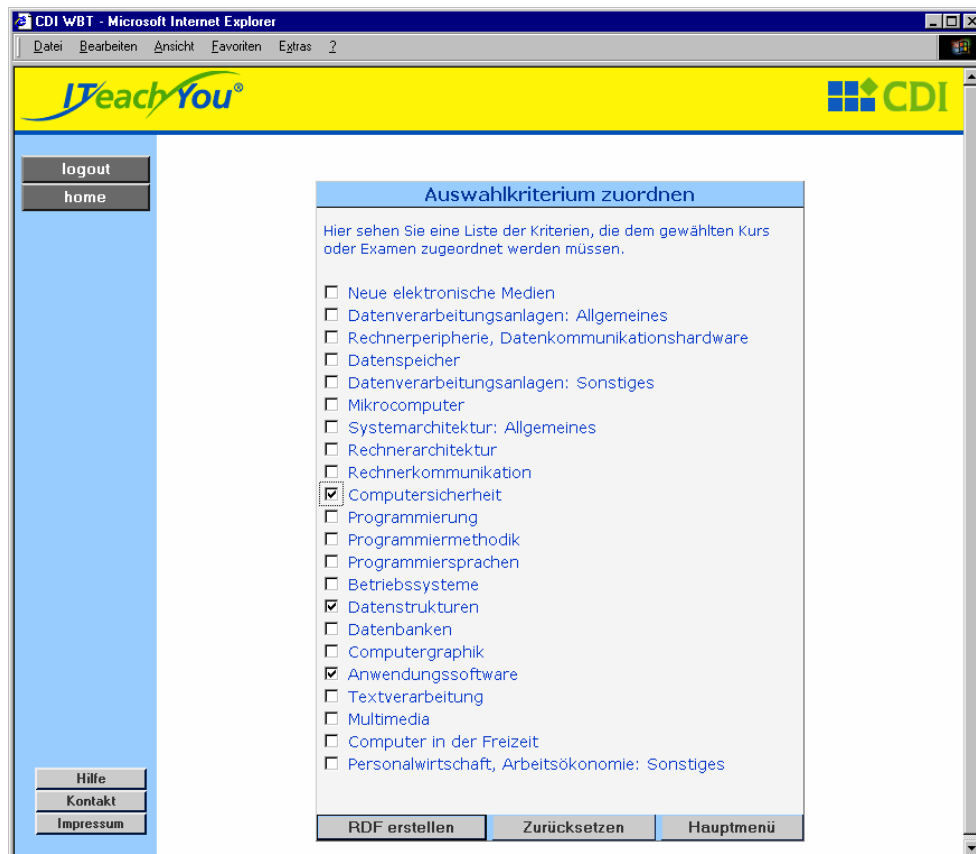


Figure 7.6: Dutch Basic Classification (Codes)

You create the XML-File by pressing the button “RDF erstellen”. Then you will get the feedback with the filename in the following window (Figure 7.7).

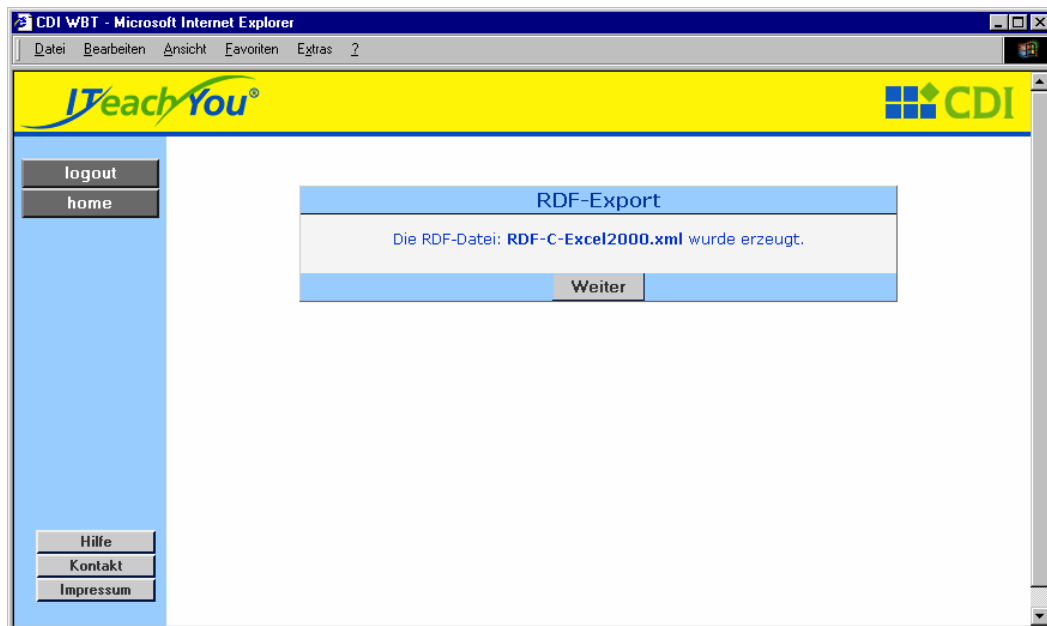


Figure 7.7: Feedback of file creation

You get back to the Main menu of "RDF-Export" by pressing "Weiter" (Figure 7.8).

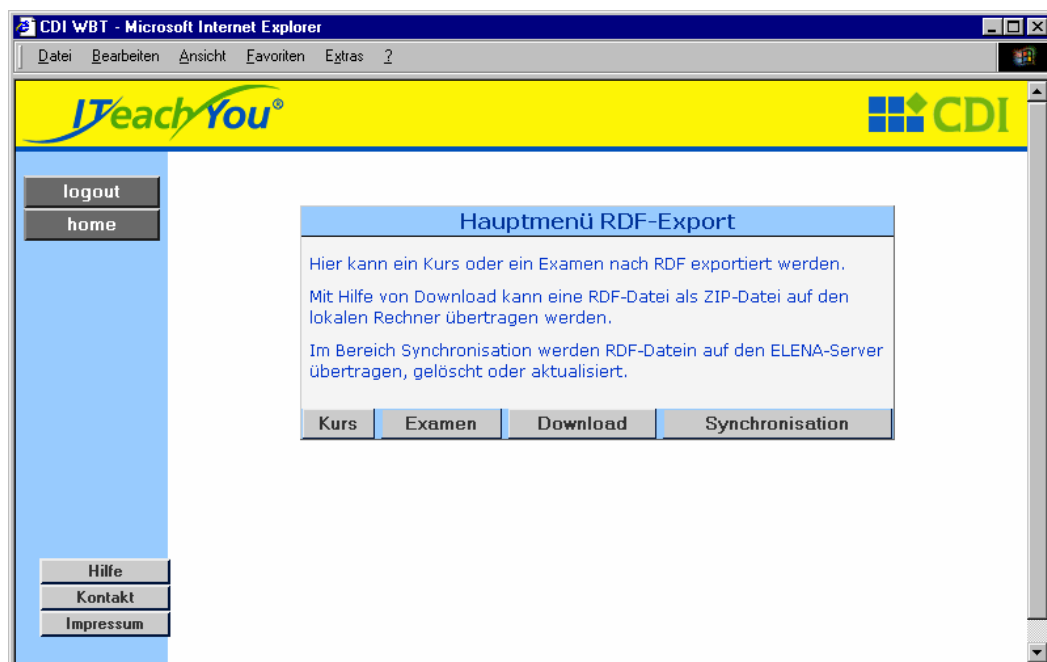


Figure 7.8: Main menu of "RDF-Export"

After having created XML-files you may press the button "Synchronisation". You can now decide if you want to insert, to update or to delete metadata on the UBP-server (Figure 7.9).

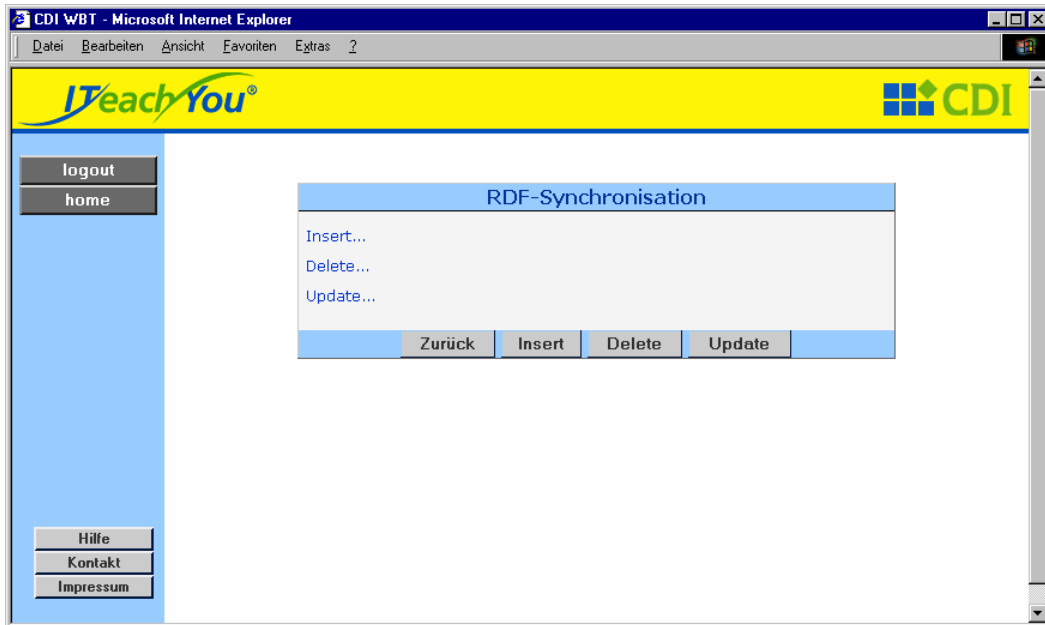


Figure 7.9: Main menu of “RDF-Synchronisation”

7.1.2 ULI Campus (*Universitärer Lehrverbund Informatik*)

The ULI project [11] (University teaching network for computer science), promoted by the “future investment program” of the BMBF is funded by the German government, and tries to establish an exchange of course material, courses and certificates in the area of computer science. Eleven German universities with eighteen different professors have agreed to exchange their courses and to allow students from one university to attend courses at another university, using advanced e-learning technologies. The project's aim is a (partial) virtualisation of Computer Science studies for the following reasons:

- For a growing number of students, a full time attendance study is difficult or impossible to assist due to family or occupation reasons. A partially virtual curriculum with courses that are not dependent on time and place can enable these students to participate in an up to standard Computer science study.
- For students from the Fernuniversität (distance learning university) Hagen, this cooperation offers the possibility to make use of the other universities' variety of offers and take part in their courses.

ULI offers the following:

- Each student of any partner university involved can register in ULI and use itself the training offer of all partners.
- For registered students the participation in the ULI courses is free of charge; contrary to regular distance courses at the Fernuni-Hagen.

- The performance records of the ULI courses are recognized at the universities of the participating partners, however at the moment this is still the most crucial part: An average of 4 ULI-courses are recognized by the typical ULI partner university.
- All ULI courses are offered virtually in the internet and require no regular physical presence

For more information about the project, we refer to the ULI-project homepage (<http://www.uli-campus.de>).

Artefacts descriptions at ULI

The ULI system provides courses focused on the computer science field. This descriptions are coded into RDF files with all metadata needed to structure the whole course. A course is composed of many learning resources, but the metadata/RDF files include the information and description of the whole course together with the description of each individual resource (see Appendix E for more examples of artefact descriptions provided in RDF).

For annotating, these resources have been defined as best-practice subset of 15 elements which are summarized in the following table, using the categories defined in LOM:

1. General	1.2 Title	dc:title
	1.3 Language	dc:language
	1.4 Description	dc:description
2. Lifecycle	2.3 Contribute	dc:creator with a lom:entity and the author in vCard format "name surname" dcq:created with the date in W3C format
6. Rights	6.3 Description	dc:rights
7. Relation		dcq:hasFormat dcq:isFormatOf dcq:hasPart dcq:isPartOf dcq:hasVersion dcq:isVersionOf dcq:requires dcq:isRequiredBy
9. Classification		dc:subject for content classification.

		This attribute links to an entry in a hierarchical ontology, that is an instance of lom_cls:Taxonomy
--	--	--

The classification of the content of a learning resource is made by means of a hierarchical ontology, in order to specify both sub- and super-topics. To be more general, the ACM CCS taxonomy (see <http://www.acm.org/class/1998/>) has been used. The ACM Computer Classification system has been used by the Association for Computer Machinery for several decades to classify scientific publications in the field of computer science. On the basic level, we find 11 nodes that split up in two more levels. Part of the classification hierarchy (2 levels) is reproduced in the following:

- A. General Literature
 - A.0 GENERAL
 - A.1 INTRODUCTORY AND SURVEY
 - A.2 REFERENCE (e.g., dictionaries, encyclopedias, glossaries)
 - A.m MISCELLANEOUS
- B. Hardware
 - B.0 GENERAL
 - B.1 CONTROL STRUCTURES AND MICROPROGRAMMING (D.3.2)
 - B.2 ARITHMETIC AND LOGIC STRUCTURES
 - B.3 MEMORY STRUCTURES
 - B.4 INPUT/OUTPUT AND DATA COMMUNICATIONS
 - B.5 REGISTER-TRANSFER-LEVEL IMPLEMENTATION
 - B.6 LOGIC DESIGN
 - B.7 INTEGRATED CIRCUITS
 - B.8 PERFORMANCE AND RELIABILITY (C.4)
 - B.m MISCELLANEOUS
- C. Computer Systems Organization
 - C.0 GENERAL
 - C.1 PROCESSOR ARCHITECTURES

- C.2 COMPUTER-COMMUNICATION NETWORKS
- C.3 SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS (J.7)
- C.4 PERFORMANCE OF SYSTEMS
- C.5 COMPUTER SYSTEM IMPLEMENTATION
- C.m MISCELLANEOUS
- D. Software
 - D.0 GENERAL
 - D.1 PROGRAMMING TECHNIQUES (E)
 - D.2 SOFTWARE ENGINEERING (K.6.3)
 - D.3 PROGRAMMING LANGUAGES
 - D.4 OPERATING SYSTEMS (C)
 - D.m MISCELLANEOUS
- E. Data
 - E.0 GENERAL
 - E.1 DATA STRUCTURES
 - E.2 DATA STORAGE REPRESENTATIONS
 - E.3 DATA ENCRYPTION
 - E.4 CODING AND INFORMATION THEORY (H.1.1)
 - E.5 FILES (D.4.3, F.2.2, H.2)
 - E.m MISCELLANEOUS
- F. Theory of Computation
 - F.0 GENERAL
 - F.1 COMPUTATION BY ABSTRACT DEVICES
 - F.2 ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY (B.6, B.7, F.1.3)
 - F.3 LOGICS AND MEANINGS OF PROGRAMS
 - F.4 MATHEMATICAL LOGIC AND FORMAL LANGUAGES

- F.m MISCELLANEOUS
- G. Mathematics of Computing
 - G.0 GENERAL
 - G.1 NUMERICAL ANALYSIS
 - G.2 DISCRETE MATHEMATICS
 - G.3 PROBABILITY AND STATISTICS
 - G.4 MATHEMATICAL SOFTWARE
 - G.m MISCELLANEOUS
- H. Information Systems
 - H.0 GENERAL
 - H.1 MODELS AND PRINCIPLES
 - H.2 DATABASE MANAGEMENT (E.5)
 - H.3 INFORMATION STORAGE AND RETRIEVAL
 - H.4 INFORMATION SYSTEMS APPLICATIONS
 - H.5 INFORMATION INTERFACES AND PRESENTATION (e.g., HCI) (I.7)
 - H.m MISCELLANEOUS
- I. Computing Methodologies
 - I.0 GENERAL
 - I.1 SYMBOLIC AND ALGEBRAIC MANIPULATION
 - I.2 ARTIFICIAL INTELLIGENCE
 - I.3 COMPUTER GRAPHICS
 - I.4 IMAGE PROCESSING AND COMPUTER VISION
 - I.5 PATTERN RECOGNITION
 - I.6 SIMULATION AND MODELING (G.3)
 - I.7 DOCUMENT AND TEXT PROCESSING (H.4, H.5)
 - I.m MISCELLANEOUS

- J. Computer Applications
 - J.0 GENERAL
 - J.1 ADMINISTRATIVE DATA PROCESSING
 - J.2 PHYSICAL SCIENCES AND ENGINEERING
 - J.3 LIFE AND MEDICAL SCIENCES
 - J.4 SOCIAL AND BEHAVIORAL SCIENCES
 - J.5 ARTS AND HUMANITIES
 - J.6 COMPUTER-AIDED ENGINEERING
 - J.7 COMPUTERS IN OTHER SYSTEMS (C.3)
 - J.m MISCELLANEOUS

- K. Computing Milieux
 - K.0 GENERAL
 - K.1 THE COMPUTER INDUSTRY
 - K.2 HISTORY OF COMPUTING
 - K.3 COMPUTERS AND EDUCATION
 - K.4 COMPUTERS AND SOCIETY
 - K.5 LEGAL ASPECTS OF COMPUTING
 - K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS
 - K.7 THE COMPUTING PROFESSION
 - K.8 PERSONAL COMPUTING
 - K.m MISCELLANEOUS

The classification has a fourth level containing unordered keywords, thus including about 1600 entries on all four levels.

In the context of the ULI project this classification turned out to fit very well, because it covers the whole field of computer science, just as the different ULI courses cover the whole discipline.

This link points to the RDF schema used for the ACM classification in ULI project:
http://www.kbs.uni-hannover.de/Uli/ACM_CCS.rdf

Artefacts provided

In this winter semester 15 courses are held as ULI courses. The total number of ULI-courses will be around 40. All of them could be provided as artefacts to the smart learning space. A list of all courses available can be retrieved from <http://www.uli-campus.de/german/kursprogramm3.html>.

Roles

In the ULI-project we differentiate between the following roles:

Roles	Notes
Student/Learner	
Assistant/Tutor	Understood as a contact, not as a trainer. Due to the nature of those courses (university courses), a trainer to advise or council is not needed
Professor	
Author	

Filtering and Mappings

Only learning resources, having the computer science classification assigned, are relevant for the smart learning space. All of the ULI courses are focused on computer science and classified into ACM classification taxonomy. Also, all the courses are thought to allow students to attend courses by means of e-learning technologies. For this reason all the courses in ULI-project are self-studying. No filter is needed.

As ULI-project uses LOM schema for describing resources, no mapping is required.

Implementation

ULI courses providers are running at the Information System Institute, University of Hannover (UHANN). ULI courses are designed according to the program described at <http://www.uli-campus.de/english/course3.html>. Currently there are three courses available: Artificial Intelligence – UHANN, Internet Applications – UNI Karlsruhe, Theory of Algorithms – UNI Freiburg. The courses are hosted by university sites described at <http://www.uli-campus.de/english/course3.html>.

At UHANN, ULI is based on the Edutella file-based provider together with metadata descriptions about course mentioned above. The ACM classification is used (<http://www.acm.org/class/1998/>) to classify learning resources. All RDF descriptions including the classification used are available from <http://www.kbs.uni-hannover.de/Uli/>.

7.2 Other Nodes

7.2.1 Personal Learning Assistant

In a Smart Space for Learning many types and instances of PLAs can exist. Some PLAs might be built according to the specification described in the D1.3 (Smart Learning Space Service Specification), some can be variations of this specification and others might be based on a different framework.

A Personal Learning Assistant (PLA) supports learners in searching for, selecting and contracting Learning Services. The PLA serves a learner or group of learners, who will use the PLA to search for learning services and trigger bookings. A PLA is capable of forwarding (modified) search requests to the Edutella-based Artefacts and Service network and can use the learning service repository of a learning management network as additional source for its recommendation. A PLA is capable of registering learners at courses of learning service providers using web-services. Notifications can be sent to learners in order to recommend learning services.

Query workflow

An example of the query management realized by a PLA in our Smart Space for Learning could be the following. Users can submit queries to the PLA, the PLA transforms the users' queries by taking the user profiles (topics, location) and the smart space ontology into account. The query results may be filtered and ranked according to information stored in the user profile (e.g. topic weights) and by concluding from behaviour of other learners (collaborative filtering). Learners can manage their search results, maintain a list of favourites by adding and removing search results from the list.

The processing of a user query is done according to the steps illustrated in Figure 7.10 (more information about each element could be found in the D1.3 - Smart Learning Space Service Specification). The process is controlled by the query workflow engine. The query workflow engine as an application engine is one of several workflow engines (examples of other workflow / application engines are learning service evaluation engine, contracting engine, etc.).

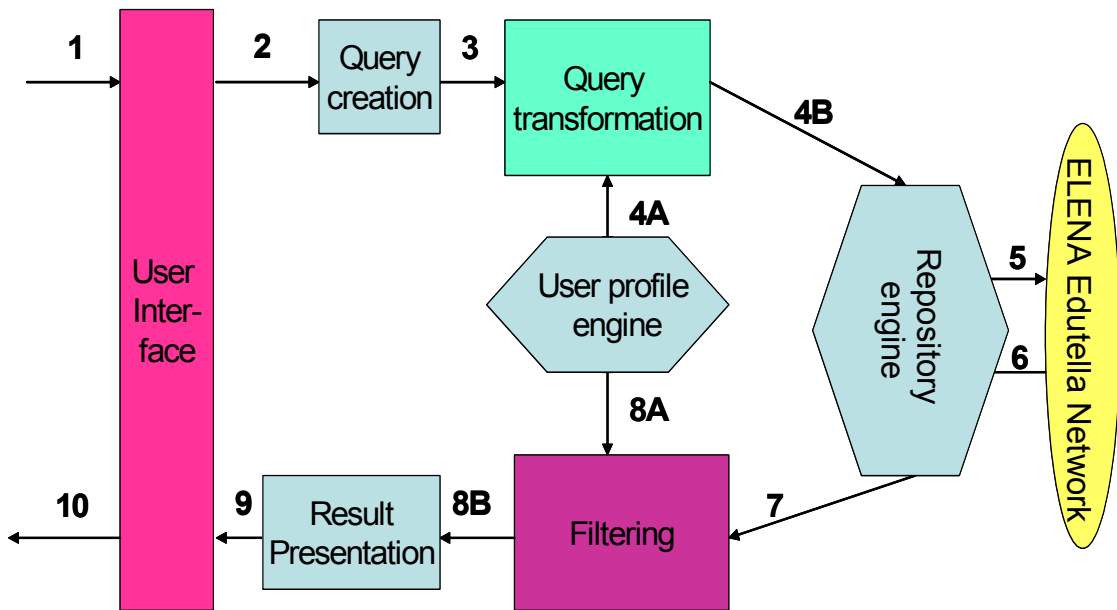


Figure 7.10: Processing of a user query

The steps can be described as follows:

1. The user enters a query via the user interface. This can happen by entering a simple text search or by selecting values from ontologies. The ontologies are not hard-coded in the user interface rather configurable. Examples for ontologies are language ontologies, subject ontologies, learning service type ontologies, ontologies for locations, etc.
2. The user input triggers the query workflow engine to call up the query creation module in order to generate a QEL query based (output parameter) on the user input (input parameter) represented through attribute-value pairs (e.g. freetext="Office Products"; title="Project Management", Language="DE")
3. The query creation module builds a QEL query, which is forwarded to the query transformation engine. A reference to the user's profile is also forwarded to the query transformation engine. The user id is necessary to find the corresponding profile.
4. The query transformation engine reads the relevant user profile. At a later stage the query transformation engine can even access distributed user profiles stored at educational nodes, which had interaction with the user in the past. Using the inference engine (which is based on TRIPLE) the query is transformed and the personalized query is sent to the repository engine (in QEL format)
5. The repository engine issues this query to the Edutella network (or possibly to the local repository)
6. The repository engine receives the response from the Edutella network (or the local repository).

7. The response is passed on to the filtering engine.
8. The filtering engine identifies which user issued from the user id provided by the workflow engine. It reads the corresponding learning profiles and/or recommendation rules.
9. The filtering engine passes the personalized results to the result presentation, which creates objects that can be handled by the user interface engine. Here, for example, a transformation from XML/RDF files to Java objects takes place.
10. The user interface engine represents the responses to the user.

7.2.2 Java client for manually operating the Synchronization interface

A Java client is available that can be used to manually synchronize resource descriptions with one of the UBP instances. This Java client is available at: <http://nm.wu-wien.ac.at/e-learning/interoperability/SynchronizationTestTool.zip>. Please follow the instructions in the file `RUN.HOWTO` which can be found in the ZIP file.

In order to synchronize an XML/RDF description file with an instance of UBP, one has to create a valid XML/RDF resource description file. A specification on the exact XML/RDF file format to use can be found in the same folder as above named "`XML_RDF_File_Format_Specs.doc`".

XML/RDF test files can be found under the following URLs:

- http://universal.infonova.at/UNIV_EXP/testEM.rdf
- http://universal.infonova.at/UNIV_EXP/testEA.rdf

7.2.3 Edutella consumer

A simple Edutella consumer can be used to query the network above. The most recent version of the "Dublin Core" consumer can be downloaded from <http://edutella.jxta.org/downloads/index.html>.

Installable software can be selected with or without a Java virtual machine. Upon first start a name for the consumer must be given. In the main window a list of providers should be displayed also showing the two UBP providers above.

By entering keywords into the Title or Description field one can search selected peers or the entire network (see Figure 7.11 below).

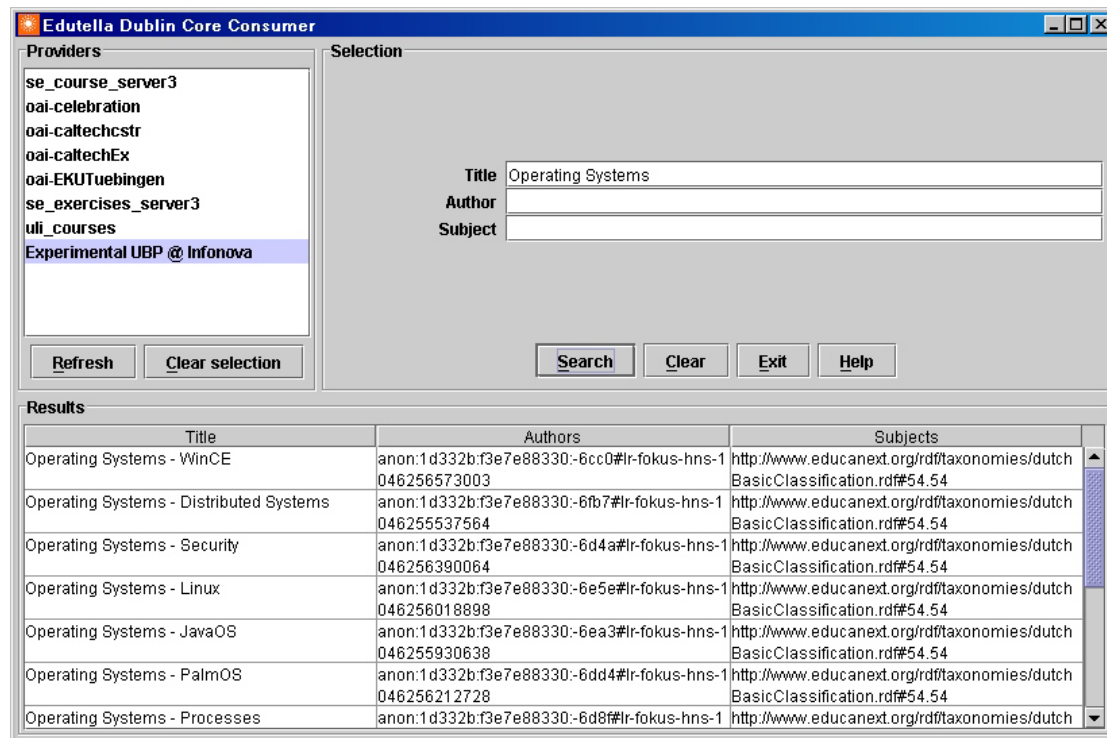


Figure 7.11: Running the Edutella Dublin Core Consumer

References

- [1] Li Gong: Project JXTA: A Technology Overview. SUN Microsystems. April 2001. Available at: <http://www.jxta.org/project/www/docs/TechOverview.pdf>.
- [2] W. Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmer, Tore Risch Edutella: A P2P Networking Infrastructure Based on RDF. 11th International World Wide Web Conference (WWW2002), Hawaii, USA, May 2002.
- [3] Project JXTA Homepage. Available at: <http://www.jxta.org/>.
- [4] SUN Microsystems. JXTA v1.0 Protocols Specification organization. 2001. Available at: <http://spec.jxta.org/v1.0/docbook/JXTAProtocols.html>.
- [5] Universal Consortium. System Interface Framework for Building an Educational Brokerage Network. Bernd Simon and Stefan Brantner editors. Technical Report. September 2002.
- [6] CDI. The Metadata-Model of the CDI WBT-Server IteachYou. White Paper. CDI, 2002.
- [7] AllWeb. Arel presentation. Allweb, 2002.
- [8] Mikael Nilson and Matthias Palmér. Conzilla – towards a concept browser. Technical Report CID-53, TRITA-NA-D9911, Department of Numerical Analysis and Computing Science, KTH, Stockholm, 1999. http://kmr.nada.kth.se/papers/ConceptualBrowsing/cid_53.pdf
- [9] R.C.Przymusinski. Every logic program has a natural stratification and an iterated least fixed-point model. In *Proceedings of the ACM Symposium on Principle of Database Systems (PODS)*, pages 11-21, 1989.
- [10] T. R. Gruber. Conceptual Analysis and Knowledge Representation. Kluwer Press, 1993.
- [11] Jan Brase, Wolfgang Nejdl. Ontologies for eLearning. Technical Report (to be published in "Handbook on Ontologies", Springer-Verlag 2003). Available at: http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2002/brase_nejdl.pdf
- [12] Nejdl, Wolfgang; Wolpers, Martin; Siberski, Wolf; Schmitz, Christoph; Schlosser, Mario; Brunkhorst, Ingo; Löser, Alexander. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. Technical Report, November 2002, accepted for Publication at *12th International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.

-
- [13] Peter Dolog and Wolfgang Nejdl: Challenges and Benefits of the Semantic Web for User Modelling. In Proc. of AH2003 workshop, 2003.
- [14] Nicola Henze and Wolfgang Nejdl: Logically Characterizing Adaptive Educational Hypermedia Systems. In Proc. of AH2003 workshop, 2003.
- [15] Peter Dolog, Rita Gavriloaie, Wolfgang Nejdl, and Jan Brase: Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments. Technical Report, November 2002, accepted for Publication at *12th International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.
- [16] Sintek, M., Decker, S.: TRIPLE—A Query, Inference, and Transformation Language for the Semantic Web. In *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, June 2002.
- [17] Kifer, M., Lausen, G., Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages, In *Journal of the ACM*, Jul. 1995., vol 42, pp. 741-843.
- [18] Decker, S., Brickley, D., Saarela J., Angele, J.: A query and inference service for RDF. In *The Query Languages Workshop, QL'98*, WorldWideWeb Consortium (W3C), Boston, USA, 1998., <http://www.w3.org/TandS/QL/QL98/>.
- [19] Horrocks, I.: The FaCT System. 2001., <http://www.cs.man.ac.uk/~horrocks/FaCT/>.

Appendix A: RDF-QEL levels

RDF-QEL-1

The RDF-QEL-1 syntax design is driven by its simplicity and readability: Following a QBE (Query By Example) paradigm queries are represented using ordinary RDF graphs having exactly the same structure as the answer graph, with additional annotations to denote variables and constraints on them. Any RDF graph query can be interpreted as a logical (conjunctive) formula that is to be proven from a knowledge base.

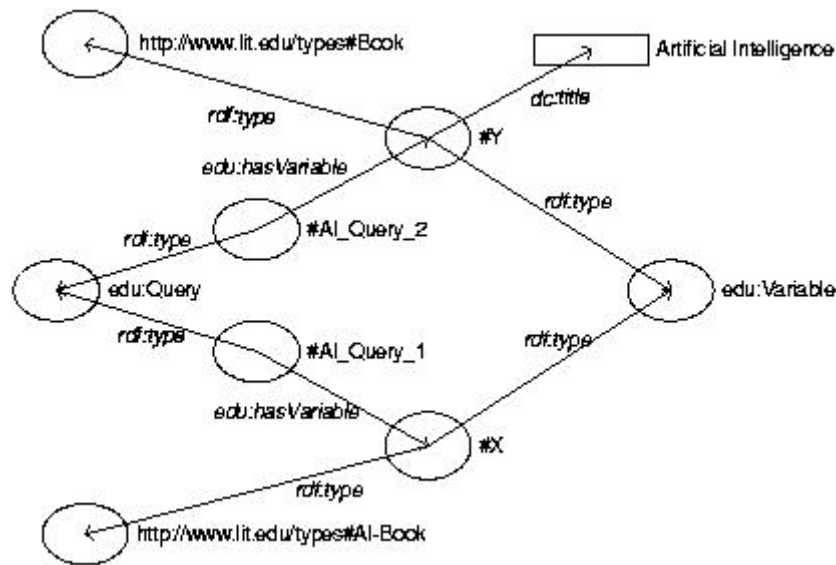


Figure A.1 Example Query in RDF-QEL-1, Unreified Format

Since disjunction cannot be expressed in RDF-QEL-1 our example query has to be split into two separate sub queries (Figure A.1).

```
<edu:QEL1Query rdf:ID="AI_Query_1">
  <edu:hasVariable rdf:resource="#X"/>
</edu:QEL1Query>

<edu:Variable rdf:ID="X" rdfs:label="X">
  <rdf:type rdf:resource="http://www.lit.edu/types#AIBook"/>
</edu:Variable>

<edu:QEL1Query rdf:ID="AI_Query_2">
  <edu:hasVariable rdf:resource="#Y"/>
</edu:QEL1Query>

<edu:Variable rdf:ID="Y" rdfs:label="X">
  <rdf:type rdf:resource="http://www.lit.edu/types#Book"/>
  <dc:title>Artificial Intelligence</dc:title>
</edu:Variable>
```

RDF-QEL-2

Extending RDF-QEL-1 with disjunction leads to RDF-QEL-2. As this language is no longer purely assertional, it cannot be expressed directly in RDF without talking *about* RDF triples in order to combine them logically. For this purpose, we utilize the RDF construct called *reification*. Reifying an RDF statement involves creating a model of the RDF triple in the form of an RDF resource of type Statement. This resource has as properties the subject, the predicate and the object of the modeled RDF triple. Such reified statements are the building blocks for each query and can, in RDF-QEL-2, be linked together by an AND-OR tree.

In RDF-QEL-2 the example query reads like

```
<edu:Variable rdf:about="#X" rdfs:label="X"/>
<edu:And rdf:about="#andbagID">
  <rdf:_2 rdf:resource="#st2"/>
  <rdf:_1 rdf:resource="#st3"/>
</edu:And>

<edu:Or rdf:about="#orbagID">
  <rdf:_1 rdf:resource="#andbagID"/>
  <rdf:_2 rdf:resource="#st1"/>
</edu:Or>

<edu:QueryStatement rdf:about="#st1">
  <rdf:subject rdf:resource="#X"/>
  <rdf:object rdf:resource="http://www.lit.edu/types#AIBook"/>
  <rdf:predicate rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"/>
</edu:QueryStatement>

<edu:QueryStatement rdf:about="#st2">
  <rdf:subject rdf:resource="#X"/>
  <rdf:object rdf:resource="http://www.lit.edu/types#Book"/>
  <rdf:predicate rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"/>
</edu:QueryStatement>

<edu:QueryStatement rdf:about="#st3">
  <rdf:object>Artificial Intelligence</rdf:object>
  <rdf:subject rdf:resource="#X"/>
  <rdf:predicate rdf:resource="http://purl.org/dc/elements/1.1/title"/>
</edu:QueryStatement>
```

The advantage of the RDF-QEL-2 form is that queries can easily be visualized using a query graph. The Conzilla query interface [8] is based on a subset of UML, using the UML specialization relationship for logical OR and the UML aggregation relationship for logical AND. As shown in figure A.2, our current prototype uses a graph-view, which is displayed as ordinary RDF with the exception that the triplets searched for (which are reified in RDF-QEL-*i*, where $n > 1$) are displayed as dashed arrows indicating that they are searched for. The logical view is displayed as a parse tree. This is the logical combination of the primitive statements, showing which combinations that should be matched at the same time in order for the query to succeed. The connections between the different views are displayed by highlighting the corresponding parts.

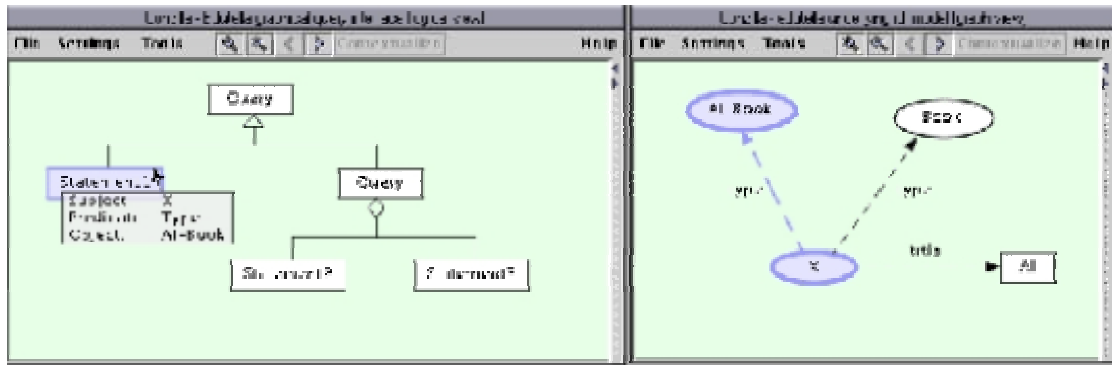


Figure A.2 Edutella Graph Query Interface

Queries can be stored and reused later, thus we can work with a library of queries that can be combined to new queries. Those queries can either be used as is or as templates, where sub-strings, numerical values, etc are filled in. Details of sub-queries can be suppressed by hiding them in detailed maps that can be presented hierarchically.

RDF-QEL-3

Going a step further, we might actually choose to skip RDF-QEL-2 in favor of RDF-QEL-3, which allows conjunction, disjunction and negation of literals. RDF-QEL-3 is essentially Datalog. Hence, the query is a set of Datalog rules, which can be encoded easily using reified statements (as for RDF-QEL-2), introducing additional constructs for negation and implication. As long as queries are non-recursive this approach is relationally complete.

The example query expressed in RDF-QEL-3 resembles the internal Datalog model described above.

```
<edu:QEL3Query rdf:ID="AI_Book_Query">
  <edu:hasQueryLiteral rdf:resource="st0"/>
  <edu:hasRule rdf:resource="r1"/>
  <edu:hasRule rdf:resource="r2"/>
</edu:QEL3Query>

<edu:Variable rdf:ID="X" rdfs:label="X"/>

<edu:Rule rdf:ID="r1">
  <edu:hasHead rdf:resource="st0"/>
  <edu:hasBody rdf:resource="st2"/>
  <edu:hasBody rdf:resource="st3"/>
</edu:Rule>

<edu:Rule rdf:ID="r2">
  <edu:hasHead rdf:resource="st0"/>
  <edu:hasBody rdf:resource="st1"/>
</edu:Rule>
```

```

<edu:QueryStatement rdf:ID="st0">
  <edu:predicate rdf:resource="aibook"/>
  <edu:arguments>
    <rdf:Seq>
      <rdf:_1 rdf:resource="#X"/>
    </rdf:Seq>
  </edu:arguments>
</edu:QueryStatement>

<edu:QueryStatement rdf:ID="st1">
  <rdf:subject rdf:resource="#X"/>
  <rdf:object rdf:resource="http://www.lit.edu/types#AIBook"/>
  <rdf:predicate rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"/>
</edu:QueryStatement>

<edu:QueryStatement rdf:ID="st2">
  <rdf:subject rdf:resource="#X"/>
  <rdf:object rdf:resource="http://www.lit.edu/types#Book"/>
  <rdf:predicate rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"/>
</edu:QueryStatement>

<edu:QueryStatement rdf:ID="st3">
  <rdf:object>Artificial Intelligence</rdf:object>
  <rdf:subject rdf:resource="#X"/>
  <rdf:predicate rdf:resource="http://purl.org/dc/elements/1.1/title"/>
</edu:QueryStatement>

```

Further RDF-QEL-i Levels

- **RDF-QEL-4:** RDF-QEL-4 allows recursion to express transitive closure and linear recursive query definitions, compatible with the SQL3 capabilities. So a relational query engine with full conformance to the SQL3 standard will be able to support the RDF-QEL-4 query level.
- **RDF-QEL-5:** Further levels allow arbitrary recursive definitions in stratified or dynamically stratified Datalog, guaranteeing one single minimal model and thus unambiguous query results ([9]).¹

RDF-QEL-i-A: Support for the usual aggregation functions as defined by SQL2 (e.g. COUNT, AVG, MIN, MAX) will be denoted by appending “-A” to the query language level, i.e. RDF-QEL-1-A, RDF-QEL-2-A, etc. RDF-QEL-i-A includes these aggregation functions as edu:count, edu:avg, edu:min, etc. Additional

¹ Technically, when using negation, recursion and the ternary representation of statements, static stratification can never be guaranteed (because we only use one ternary predicate “s(S,P,O)”), so we have to rely on dynamic stratification (which depends on the actual instantiation of literals) or switch to well-founded semantics.

“foreign” functions like edu:substring etc. to be used in conditions might be useful as well, but have not been included yet in RDF-QEL-i-A.

Appendix B: Simple Edutella consumer and provider

This chapter describes a simple implementation of an Edutella consumer and provider. This is work in progress and thus subject to changes. The structural model of implementation for a simple Edutella provider is depicted in fig. B.1.

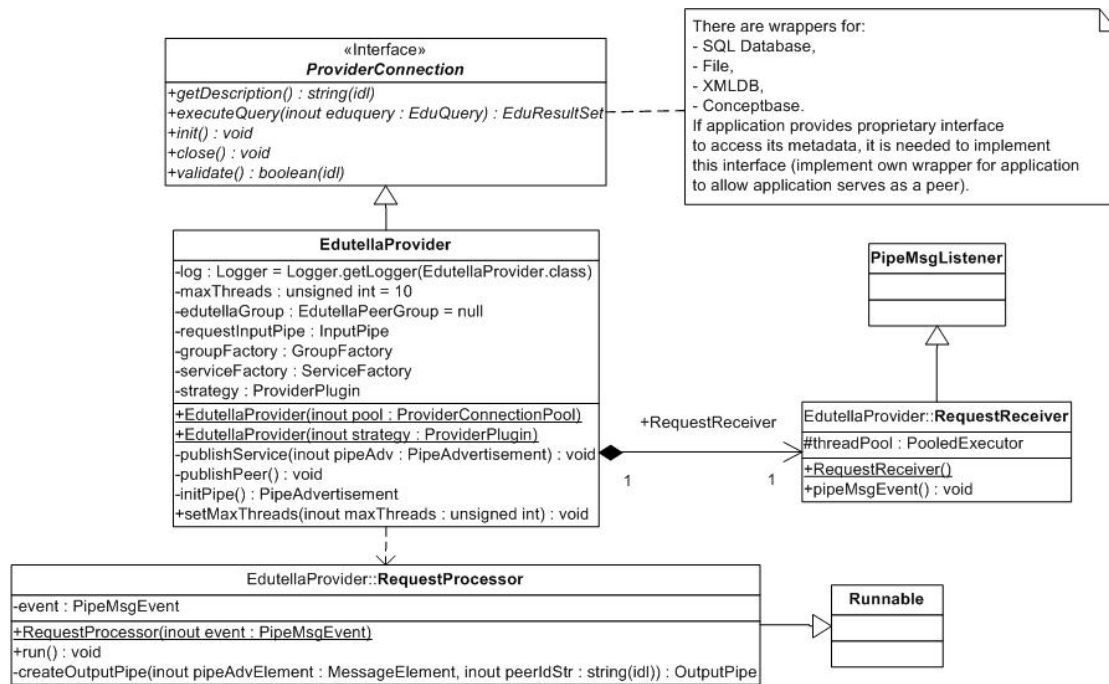


Figure B.1 Simple implementation of ProviderConnection interface

An Edutella peer component provides a query service interface and query results interface. The query interface is used by the service provider, but can be exported directly to a user. The query interface can also be used by an LMS as user interface provider. In the current implementation of QueryService interface the query results are expressed in RDF. The generation of appropriate results for a user will be the aim of the user interface providers (LMS). Structural model of a simple implementation of the QueryService interface is depicted in fig. B.2.

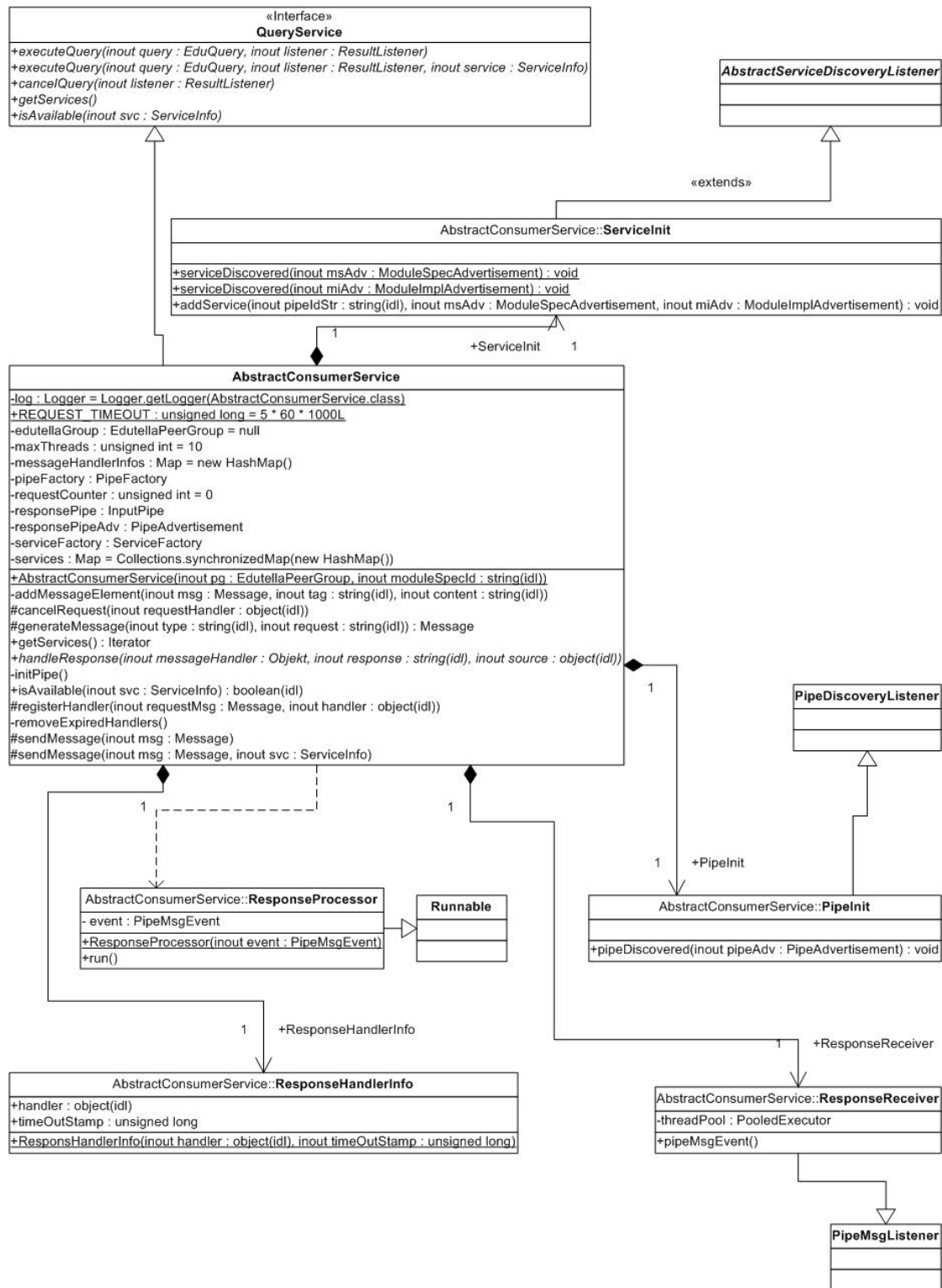


Figure B.2 Simple implementation of QueryService interface

The interface for peers searching and for broadcasting query need not to be provided – it could be and it is now internal functionality of Edutella provider. Query results interface can be used by LMS and by service provider. Query results are provided in RDF.

Appendix C: Interface specification for Edutella provider and consumer

The ProviderConnection interface specification:

```
public interface ProviderConnection extends PooledConnection {
    /**
     * returns a provider description
     *
     * @return human readable description of the provider
     */
    public String getDescription();

    /**
     * calculates a result for a query.
     *
     * @param query the query to process
     * @return result set containing the result(s)
     */
    public EduResultSet executeQuery(EduQuery eduquery); // handle
query

    /**
     * initializes the connection.
     *
     */
    public void init();

    /**
     * closes the connection.
     *
     */
    public void close();

    /**
     * tests the connection.
     *
     * @return true, if connection is still usable
     */
    public boolean validate();
}
```

The QueryService interface specification:

```
public interface QueryService {
    /**
     * distributes a query to all discovered providers.
     * Results are sent to the ResultListener
     * provided by the caller.
     *
     * @param query the query to execute
     * @param listener results are sent to this ResultListener
     */
    public void executeQuery(EduQuery query, ResultListener listener);

    /**
     * distributes a query to a specific provider. Results are sent
```

```
* to the ResultListener provided by the caller.
*
* @param query the query to execute
* @param listener results are sent to this ResultListener
* @param service information on the peer to be queried
*/
public void executeQuery(
    EduQuery query,
    ResultListener listener,
    ServiceInfo service);

/**
 * cancels the query which was posed using this listener.
 * @param listener
 */
public void cancelQuery(ResultListener listener);

/**
 * returns a list of all available services
 *
 */
public Iterator getServices();

/**
 * checks if a specific service is still available
 */
public boolean isAvailable(ServiceInfo svc);

}
```

Appendix D: Learning Resource Descriptions at Educanext

Learning Resource Description: Educational Activity

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcq="http://dublincore.org/2000/03/13/dcq#"
  xmlns:lom="http://www.imsproject.org/rdf/imsmd_rootv1p2#"
  xmlns:lom_gen="http://www.imsproject.org/rdf/imsmd_generalv1p2#"
  xmlns:lom_meta="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#"
  xmlns:lom_edu="http://www.imsproject.org/rdf/imsmd_educationalv1p2#"
  xmlns:lom_life="http://www.imsproject.org/rdf/imsmd_lifecyclev1p2#"
  xmlns:univ_lr="http://tristan.wu-wien.ac.at/universal/rdf/univ-lr#"
  xmlns:catalog="http://tristan.wu-wien.ac.at/universal/rdf/catalogs#">

  <!-- ===== Metametadata. ===== -->

  <rdf:Description about="">

    <!-- Language used in this metadata-file (when no langstring is
    defined explicitly)-->

    <dc:language rdf:resource="http://tristan.wu-wien.ac.at/universal/rdf/lang.rdf#en"/>

    <!-- Creator of this metadata-file = spoken in UNIVERSAL terms:
    provider -->

    <dc:creator rdf:resource="urn:univ:us-WUW-tenzi"/>
    <dcq:created>
      <dcq:W3CDTF>
        <rdf:value>2002-04-20</rdf:value>
      </dcq:W3CDTF>
    </dcq:created>

    <lom_gen:catalogentry>
      <catalog:universal>
        <rdf:value>lr-wuw-tenzi-1</rdf:value>
      </catalog:universal>
    </lom_gen:catalogentry>

    <lom_meta:metadatascheme
    rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#LO
    Mv1.0"/>
      <lom_meta:metadatascheme
    rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#AR
    IADNEv3"/>
        <lom_meta:metadatascheme
    rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#IM
    SMD_1.2"/>
          <lom_meta:metadatascheme rdf:resource="http://tristan.wu-
    wien.ac.at/universal/rdf/universal_metadata#UNIVERSAL_1.0"/>
```

```

</rdf:Description>

<!-- ===== Learning Resource metadata ===== -->

<rdf:Description rdf:ID="urn:univ:lr-WUW-tenzi-1">

  <dc:title>
    <lom:LangString rdf:ID="title">
      <rdf:value rdf:parseType="Resource">
        <rdf:value>
          The Emerging Paradigm for Electronic Commerce
        </rdf:value>
        <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
      </rdf:value>
      <lom:translation rdf:parseType="Resource">
        <rdf:value>
          Die Aufstrebenden Paradigmen im Electronic Commerce
        </rdf:value>
        <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
      </lom:translation>
    </lom:LangString>
  </dc:title>

  <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
  <dc:description>
    <lom:LangString rdf:ID="description">
      <rdf:value rdf:parseType="Resource">
        <rdf:value>

The last couple of years have seen incredible activity around the
development of technologies for businesses to communicate with each
other over the Internet - the area known as B-2-B electronic
commerce. These efforts, including industry-wide efforts, such as
RosettaNet and the UNsponsored ebXML, as well as more company
specific efforts, such as Commerce One's xCBL, have all centred
around the use of XML as a means of describing business messages.
The new paradigm for business is the exchange of XML-based messages
among businesses organized into electronic marketplaces for the
purpose of conducting business in ever more complex ways. The first
couple of generations of these supported fairly simple transactions,
but that will soon change. he most interesting new event in the XML
world is the arrival of the official XML Schemalanguage from the W3C.
This represents a significant advance over the current DTDs as a
means of expressing complex data relationships and other constraints,
and will lead to the next generation of marketplaces, where
businesses will take advantage of sophisticated services. Some of us
may see echoes of distributed object/agent technologies here. I would
not disabuse anyone of this notion - XML is ideal for specifying
agent communication languages, and businesses generally behave as
loosely coupled autonomous agents. This talk will discuss the
evolving nature of marketplaces and the role of XML in particular in
supporting this evolution.

        </rdf:value>
      </lom:LangString>
    </dc:description>
  </dc:language>
  <lom:translation rdf:parseType="Resource">
    <rdf:value>
      Die deutsche Übersetzung der obigen Beschreibung.
    </rdf:value>
  </lom:translation>
</rdf:Description>

```

```

    <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
  </lom:translation>
</lom:LangString>
</dc:description>

<univ:educationalObjectives>
  <lom:LangString rdf:ID="description">
    <rdf:value rdf:parseType="Resource">
      <rdf:value>
        Description of the educational objectives of this learning
resource.
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
    </rdf:value>
    <lom:translation rdf:parseType="Resource">
      <rdf:value>
        Beschreibung der pädagogischen Ziele bzw. des pädagogischen
Aufbaus dieser Lernressource.
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
    </lom:translation>
  </lom:LangString>
</univ:educationalObjectives>

<univ_lr:instructionalDesign rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ_lr#CollaborativeLearning"/>
<univ_lr:locationOfAdditionalInformation
rdf:resource="http://tristan.wu-
wien.ac.at/lectureseries/LectureSeries2.htm#mf"/>

<univ_lr:curriculum>
  <lom:LangString rdf:ID="curriculum">
    <rdf:value rdf:parseType="Resource">
      <rdf:value>
        University second level on WU-Wien -
        Department for Commercial Information Technology
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
    </rdf:value>
  </lom:LangString>
</univ_lr:curriculum>

<lom_edu:typicallearningtime>
  <lom:ISO8601>
    <rdf:value>PT2H</rdf:value>
  </lom:ISO8601>
</lom_edu:typicallearningtime>

<univ_lr:prerequisite>
  <lom:LangString rdf:ID="prerequisites">
    <rdf:value rdf:parseType="Resource">
      <rdf:value>
        Some basics about HTML, XML (and RDF)
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
    </rdf:value>
  </lom:LangString>
</univ_lr:prerequisite>

```

```

<univ_lr:learningResourceType rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lr#EducationalMaterial"/>
<univ_lr:mediaType rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lr#VideoConference"/>
<univ_lr:educationalActivityType rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ_lr#Lecture">

<lom_edu:context>higher education</lom_edu:context>
<univ_lr:ects>2</univ_lr:ects>

<!-- The Author of this Resource -->
<!-- us-1 ... Matthew Fuchs -->
<dc:creator rdf:resource="urn:univ:us-1"/>
<dcq:created>
  <dcq:W3CDTF>
    <rdf:value>2000-05-07</rdf:value>
  </dcq:W3CDTF>
</dcq:created>

<!-- there are two contentprovider -->
<lom_life:contentprovider rdf:parseType="Resource">
  <rdf:value>
    <rdf:Seq>
      <rdf:li rdf:resource="urn:univ:us-wuw-bsimon"/>
      <rdf:li rdf:resource="urn:univ:us-wuw-tenzi"/>
    </rdf:Seq>
  </rdf:value>
  <dc:date>
    <dcq:W3CDTF>
      <rdf:value>2001-05-08</rdf:value>
    </dcq:W3CDTF>
  </dc:date>
</lom_life:contentprovider>

<dc:subject resource="http://tristan.wu-
wien.ac.at/universal/rdf/taxonomies/tx-1-en#2.1"/>
<dc:subject resource="http://tristan.wu-
wien.ac.at/universal/rdf/taxonomies/tx-10-en#3.2.1"/>

<univ_lr:annotation rdf:resource="urn:univ:an-wuw-bsimon-1">
<univ_lr:offer rdf:resource="urn:univ:of-wuw-tenzi-1"/>

<!-- ===== Access Information ===== -->

<univ_lr:deliverySystem rdf:resource="urn:univ:ds-isabel6"/>

<lom_tech:location rdf:resource="http://isabel.dit.upm/someurl"/>
<lom_life:version>1.0</lom_life:version>

<univ_tech:requirement rdf:parseType="Resource">
  <univ_tech:type rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lo#System"/>
  <univ_tech:name rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lo#Linux"/>
</univ_tech:requirement>

<!-- ===== Scheduling because it is an Educational Activity ===== -->

<dcq:available>
  <dcq:Period rdf:ID="Period_1">
    <dcq:start>
      <dcq:W3CDTF>
        <rdf:value>2002-05-27T19:00+01:00</rdf:value>
      </dcq:W3CDTF>
    </dcq:start>
  </dcq:Period>
</dcq:available>

```

```

    </dcq:W3CDTF>
    </dcq:start>
  <dcq:end>
    <dcq:W3CDTF>
    <rdf:value>2002-05-27T20:30+01:00</rdf:value>
    </dcq:W3CDTF>
  </dcq:end>
</dcq:Period>
</dcq:available>
</rdf:Description>
</rdf:RDF>

```

Learning Resource Description: Educational Material

```

<?xml version='1.0'?>
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcq="http://dublincore.org/2000/03/13/dcq#"
  xmlns:lom="http://www.imsproject.org/rdf/imsmd_rootv1p2#"
  xmlns:lom_gen="http://www.imsproject.org/rdf/imsmd_generalv1p2#"
  xmlns:lom_meta="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#"
  xmlns:lom_edu="http://www.imsproject.org/rdf/imsmd_educationalv1p2#"
  xmlns:lom_life="http://www.imsproject.org/rdf/imsmd_lifecyclev1p2#"
  xmlns:univ_lr="http://tristan.wu-wien.ac.at/universal/rdf/univ-lr#"
  xmlns:catalog="http://tristan.wu-wien.ac.at/universal/rdf/catalogs#">

  <!-- ===== Metametadata. ===== -->

  <rdf:Description about="">

    <!-- Language used in this metadata-file (when no langstring is
    defined explicitly)-->

    <dc:language rdf:resource="http://tristan.wu-wien.ac.at/universal/rdf/lang.rdf#en"/>

    <!-- Creator of this metadata-file = spoken in UNIVERSAL terms:
    provider -->

    <dc:creator rdf:resource="urn:univ:us-WUW-tenzi"/>
    <dcq:created>
      <dcq:W3CDTF>
      <rdf:value>2002-04-20</rdf:value>
      </dcq:W3CDTF>
    </dcq:created>

    <lom_gen:catalogentry>
      <catalog:universal>
      <rdf:value>lr-wuw-tenzi-1</rdf:value>
      </catalog:universal>
    </lom_gen:catalogentry>

    <lom_meta:metadatascheme
    rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#LO
    Mv1.0"/>
    <lom_meta:metadatascheme
    rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadatav1p2#AR
    IADNEv3"/>

```

```

    <lom_meta:metadatascheme
rdf:resource="http://www.imsproject.org/rdf/imsmd_metametadav1p2#IM
SMD_1.2"/>
    <lom_meta:metadatascheme rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/universal_metadata#UNIVERSAL_1.0"/>

```

```

</rdf:Description>

```

```

<!-- ===== Learning Resource metadata ===== -->

```

```

<rdf:Description rdf:ID="urn:univ:lr-WUW-tenzi-1">

  <dc:title>
    <lom:LangString rdf:ID="title">
      <rdf:value rdf:parseType="Resource">
        <rdf:value>
          The Emerging Paradigm for Electronic Commerce
        </rdf:value>
        <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
      </rdf:value>
      <lom:translation rdf:parseType="Resource">
        <rdf:value>
          Die Aufstrebenden Paradigmen im Electronic Commerce
        </rdf:value>
        <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
      </lom:translation>
    </lom:LangString>
  </dc:title>

  <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>

  <dc:description>
    <lom:LangString rdf:ID="description">
      <rdf:value rdf:parseType="Resource">
        <rdf:value>
          The last couple of years have seen incredible activity around
          the development of technologies for businesses to communicate with
          each other over the Internet - the area known as B-2-B electronic
          commerce. These efforts, including industry-wide efforts, such as
          RosettaNet and the UNsponsored ebXML, as well as more company
          specific efforts, such as Commerce One's xCBL, have all centered
          around the use of XML as a means of describing business messages. The
          new paradigm for business is the exchange of XML-based messages among
          businesses organized into electronic marketplaces for the purpose of
          conducting business in ever more complex ways. The first couple of
          generations of these supported fairly simple transactions, but that
          will soon change.

          The most interesting new event in the XML world is the arrival
          of the official XML Schema language from the W3C. This represents a
          significant advance over the current DTDs as a means of expressing
          complex data relationships and other constraints, and will lead to
          the next generation of marketplaces, where businesses will take
          advantage of sophisticated services. Some of us may see echoes of
          distributed object/agent technologies here. I would not disabuse
          anyone of this notion - XML is ideal for specifying agent
          communication languages, and businesses generally behave as loosely
          coupled autonomous agents. This talk will discuss the evolving nature
          of marketplaces and the role of XML in particular in supporting this
          evolution.
        </rdf:value>
      </rdf:value>
    </lom:LangString>
  </dc:description>

```



```

    <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
  </rdf:value>
  <lom:translation rdf:parseType="Resource">
    <rdf:value>
      Da wäre jetzt halt die deutsche Übersetzung der obigen
      Beschreibung. Dafür ist aber keine Zeit!!! :-)
    </rdf:value>
    <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
  </lom:translation>
</lom:LangString>
</dc:description>

<univ:educationalObjectives>
  <lom:LangString rdf:ID="description">
    <rdf:value rdf:parseType="Resource">
      <rdf:value>
        Description of the educational objectives of this learning
        resource.
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
    </rdf:value>
    <lom:translation rdf:parseType="Resource">
      <rdf:value>
        Beschreibung der pädagogischen Ziele/des pädagogischen Aufbaus
        dieser Lernressource.
      </rdf:value>
      <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#de"/>
    </lom:translation>
  </lom:LangString>
</univ:educationalObjectives>

  <univ_lr:instructionalDesign rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ_lr#DirectedLearning"/>
  <univ_lr:locationOfAdditionalInformation
rdf:resource="http://tristan.wu-
wien.ac.at/lectureseries/LectureSeries2.htm#mf"/>

  <univ_lr:curriculum>
    <lom:LangString rdf:ID="curriculum">
      <rdf:value rdf:parseType="Resource">
        <rdf:value>
          University second level on WU-Wien -
          Department for Commercial Information Technology
        </rdf:value>
        <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
      </rdf:value>
    </lom:LangString>
  </univ_lr:curriculum>

  <lom_edu:typicallearningtime>
    <lom:ISO8601>
      <rdf:value>PT2H</rdf:value>
    </lom:ISO8601>
  </lom_edu:typicallearningtime>

  <univ_lr:prerequisite>
    <lom:LangString rdf:ID="prerequisites">
      <rdf:value rdf:parseType="Resource">

```

```

    <rdf:value>
      Some basics about HTML, XML (and RDF)
    </rdf:value>
    <dc:language rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/lang.rdf#en"/>
  </rdf:value>
  </lom:LangString>
</univ_lr:prerequisite>

  <univ_lr:learningResourceType rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ_lr#EducationalMaterial"/>
  <univ_lr:educationalMaterialType rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ_lr#Figure">

  <lom_edu:context>higher education</lom_edu:context>
  <univ_lr:ects>2</univ_lr:ects>

  <!-- The Author of this Resource -->
  <!-- us-1 ... Matthew Fuchs -->
  <dc:creator rdf:resource="urn:univ:us-1"/>
  <dcq:created>
    <dcq:W3CDTF>
      <rdf:value>2000-05-07</rdf:value>
    </dcq:W3CDTF>
  </dcq:created>

  <!-- there are two content providers -->
  <lom_life:contentprovider rdf:parseType="Resource">
    <rdf:value>
      <rdf:Seq>
        <rdf:li rdf:resource="urn:univ:us-wuw-bsimon"/>
        <rdf:li rdf:resource="urn:univ:us-wuw-tenzi"/>
      </rdf:Seq>
    </rdf:value>
    <dc:date>
      <dcq:W3CDTF>
        <rdf:value>2001-05-08</rdf:value>
      </dcq:W3CDTF>
    </dc:date>
  </lom_life:contentprovider>

  <dc:subject resource="http://tristan.wu-
wien.ac.at/universal/rdf/taxonomies/tx-1-en#2.1"/>
  <dc:subject resource="http://tristan.wu-
wien.ac.at/universal/rdf/taxonomies/tx-10-en#3.2.1"/>

  <univ_lr:annotation rdf:resource="urn:univ:an-wuw-bsimon-2">
  <univ_lr:offer rdf:resource="urn:univ:of-wuw-tenzi-2"/>

  <!-- ===== Access Information ===== -->

  <univ_lr:deliverySystem rdf:resource="urn:univ:ds-RealServer"/>

  <lom_tech:location rdf:resource="http://media.wu-
wien.ac.at:8080/ramgen/The_Emerging_Paradigm_for_Electronic_Commerce_
by_Matthew_Fuchs/trainer.smi"/>
  <lom_life:version>1.0</lom_life:version>

  <dcq:medium>
    <dcq:IMT>
      <rdf:value>application/smil</rdf:value>
      <rdfs:label>RealVideo</rdfs:label>
    </dcq:IMT>

```

```
</dcq:medium>

<dcq:extent>
  <lom_tech:ByteSize>
    <rdf:value>21000000</rdf:value>
  </lom_tech:ByteSize>
</dcq:extent>

  <univ_tech:requirement rdf:parseType="Resource">
    <univ_tech:type rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lo#PresentationTool"/>
    <univ_tech:name rdf:resource="http://tristan.wu-
wien.ac.at/universal/rdf/univ-lo#RealPlayer"/>
    <univ_tech:minimum>1.0</univ_tech:minimum>
    <univ_tech:maximum>8.0</univ_tech:maximum>
  </univ_tech:requirement>

</rdf:Description>
</rdf:RDF>
```

Appendix E: Learning Resource Descriptions at ULI-project

Uli Vorlesung Algorithmenthorie WS 2001 (Freiburg)

```

<rdf:RDF xml:lang="de"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcq="http://dublincore.org/2001/08/14/dcq#"
xmlns:lom_cls="http://www.imsproject.org/rdf/imsmd_classificationv1p2#"
xmlns:lom="http://ltsc.ieee.org/2002/09/lom-base#"
xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

<rdf:Description ID="Algorithmen">
<rdf:type resource="http://telemann.kbs.uni-hannover.de:3333/olr/olr_v9#Course"/>
<dc:title>Uli Vorlesung Algorithmenthorie WS 2001 (Freiburg)</dc:title>
<dc:description>In dieser Vorlesung werden Algorithmen und Datenstrukturen behandelt, die in der Vorlesung im Grundstudium überhaupt nicht oder nicht mit der erforderlichen Tiefe behandelt werden. Dazu gehören u. a.: Datenstrukturen: Skip-Listen, randomisierte Suchbäume, Splay Trees, erweiterte Suchbäume, Tries. Polynome und die schnelle Fouriertransformation. Algorithmen zur Zeichenkettenverarbeitung (Knuth Morris Pratt, Boyer Moore u. a.). Verschlüsselung und Kodierung von Texten. Approximative Verfahren zur Lösung NP-harter Probleme. Ferner werden die zum Entwurf einer breiten Palette von Algorithmen benötigten Entwurfsprinzipien herausgearbeitet und an Beispielen erläutert. Das sind das Divide-and-Conquer-Prinzip, das Greedy-Prinzip, dynamisches Programmieren, amortisierte Analyse u. a.</dc:description>
<dc:creator>
<lom:entity>
<vCard:FN>Thomas Ottman</vCard:FN>
</lom:entity>
</dc:creator>
<dcq:created>
<dcq:W3CDTF>
<rdf:value>2001-09-15</rdf:value>
</dcq:W3CDTF>
</dcq:created>

<dcq:hasPart>
<rdf:Seq>
<rdf:li resource="#Unit1"/>
<rdf:li resource="#Unit2"/>
<rdf:li resource="#Unit3"/>
<rdf:li resource="#Unit4"/>
</rdf:Seq>
</dcq:hasPart>
</rdf:Description>

<rdf:Description ID="Unit1">

```

```

<rdf:type resource="http://telemann.kbs.uni-
hannover.de:3333/olr/olr_v9#Unit"/>
<dc:title>1: Divide and Conquer</dc:title>
<dcq:isPartOf rdf:resource="#Algorithmen"/>
<dcq:hasPart>
  <rdf:Seq>
    <rdf:li rdf:resource="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at11
.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at12
.htm"/>
  </rdf:Seq>
</dcq:hasPart>
</rdf:Description>

```

```

<rdf:Description about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at11
.htm">
<dc:title>Vorlesung Algorithmtheorie WS 2001 (Freiburg) 1.1 Divide
and Conquer, Geometrisches Divide and Conquer</dc:title>
<dc:description>Als Erstes werden die verschiedenen
Entwurfsprinzipien für Algorithmen und deren Problem- und
Anwendungsbereiche kurz vorgestellt
Anschließend wird das allgemeine Entwurfsprinzip des Divide-and-
Conquer Verfahrens vorgestellt und anhand des bekannten
Sortierverfahrens
Quicksort noch einmal erklärt. Danach wird anhand von drei Beispielen
ausführlich gezeigt, wie man das Divide-and-Conquer Prinzip auch auf
geometrische Probleme anwenden kann. Für das Closest Pair-Problem,
bei dem es um die Bestimmung eines Punktepaars mit minimaler Distanz
aus einer gegebenen Menge von n Punkten geht, kann mit Hilfe eines
cleveren Merge-Verfahrens eine Laufzeit von  $O(n \log n)$  erzielt
werden.
Als Nächstes wird das Liniensegmentschnittproblem behandelt. Hier
geht es darum alle Paare sich schneidender Segmente zu bestimmen,
wobei
die gegebene Menge von n Segmenten nur aus horizontal und vertikal
orientierten Segmenten besteht. Der Divide-and-Conquer Ansatz beim
Liniensegmentschnitt-Problem führt auf ein output sensitives
Verfahren mit einer Laufzeit von  $O(n \log n + k)$ , wobei k die Anzahl
sich schneidender
Segmente ist. Zum Schluss wird noch kurz auf die Berechnung des
Voronoi-Diagramms mit Hilfe eines Divide-and-Conquer Ansatzes
eingegangen.
Bei diesem Problem geht es um die Unterteilung der Ebene in Regionen
gleicher nächster Nachbarn. Auch dieses Problem kann effizient in  $O(n \log n)$ 
Zeit gelöst werden. </dc:description>
<dc:language rdf:resource="http://www.kbs.uni-
hannover.de/Uli/lang.rdf#de"/>
<dc:subject rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#F.2.1"/>
<dcq:isPartOf rdf:resource="#Unit1"/>
</rdf:Description>

```

```

<rdf:Description about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at12
.htm">
<dc:title>Vorlesung Algorithmtheorie WS 2001 (Freiburg) 1.2 Fast
Fourier Transformation</dc:title>
<dc:description>Zunächst werden Polynome, deren
Repräsentationsmöglichkeiten und grundlegende Operationen auf diesen
eingeführt. In der

```

Koeffizientendarstellung ist das Polynomprodukt nicht sehr effizient berechenbar, daher werden die Polynome in die Punkt/Wertdarstellung transformiert,

in der dann das Polynomprodukt in $O(n)$ berechenbar ist. Anschließend wird das Ergebnis wieder zurück in die Koeffizientendarstellung transformiert.

Das Ziel ist, mit Hilfe eines Divide-and-Conquer Ansatzes ein effizientes Verfahren für diese Transformationen zu entwickeln. Dies führt schließlich auf die Fast Fourier Transformation (FFT). Der entscheidende Schritt hierbei ist, dass durch die Auswertung des Polynoms an den Potenzen der n -ten

komplexen Einheitswurzel nach dem Divide-Schritt die beiden Teilpolynome nur noch an der Hälfte der Stellen des ursprünglichen Polynoms ausgewertet

werden müssen. Dies führt zu einer Laufzeit der FFT von $O(n \log n)$. Die inverse FFT wird auf die Berechnung der FFT zurückgeführt wodurch schließlich

das Polynomprodukt insgesamt in $O(n \log n)$ berechenbar ist.

```
</dc:description>
<dc:language rdf:resource="http://www.kbs.uni-hannover.de/Uli/lang.rdf#de"/>
<dc:subject rdf:resource="http://www.kbs.uni-hannover.de/Uli/ACM_CCS.rdf#F.2.1.0"/>
<dcq:isPartOf rdf:resource="#Unit1"/>
</rdf:Description>
```

```
<rdf:Description ID="Unit2">
<rdf:type resource="http://telemann.kbs.uni-hannover.de:3333/olr/olr_v9#Unit"/>
<dc:title>2: Randomisierung</dc:title>
<dcq:isPartOf rdf:resource="#Algorithmen"/>
<dcq:hasPart>
  <rdf:Seq>
    <rdf:li rdf:resource="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at21.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at22.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at23.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at24.htm"/>
  </rdf:Seq>
</dcq:hasPart>
</rdf:Description>
```

```
<rdf:Description about="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at21.htm">
<dc:title>Vorlesung Algorithmenthorie WS 2001 (Freiburg) 2.1 Zufallszahlengeneratoren </dc:title>
<dc:description>Die Zufallsgeneratoren, die in Bibliotheken oder durch Betriebssysteme bereits gestellt werden, haben häufig sehr schlechte Eigenschaften. Es wird daher die lineare Kongruenz Methode von Lehmer vorgestellt, mit dessen Hilfe Pseudo-Zufallszahlen erzeugt werden können. Wird das Verfahren jedoch wie beschrieben implementiert, kann es zu Überläufen bei den Berechnungen kommen. Daher wird eine verbesserte Version nach Schrage vorgestellt, die dieses Problem behebt.
```

Im Anschluss daran wird kurz auf die Eigenschaften gemischt-kongruenter Generatoren und den Zufallszahlengenerator von Java eingegangen.</dc:description>

```
<dc:language rdf:resource="http://www.kbs.uni-hannover.de/Uli/lang.rdf#de"/>
<dc:subject rdf:resource="http://www.kbs.uni-hannover.de/Uli/ACM_CCS.rdf#G.3.9"/>
<dcq:isPartOf rdf:resource="#Unit2"/>
</rdf:Description>
```

```
<rdf:Description about="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmentheorie/download/at_lectures/at22.htm">
```

```
<dc:title>Vorlesung Algorithmentheorie WS 2001 (Freiburg) 2.2 Randomisiertes Quicksort, Randomisierter Primzahltest </dc:title>
```

```
<dc:description>Randomisierte Algorithmen kann man in zwei Klassen einteilen, die Las Vegas und die Monte Carlo Algorithmen.
```

Las Vegas Algorithmen sind wahrscheinlich schnell und immer korrekt. Zu dieser Klasse wird als Beispiel das Randomisierte Quicksort vorgestellt. Durch eine randomisierte Wahl des Pivotelementes erhält man als Erwartungswert für die Laufzeit $O(n \log n)$.

Monte Carlo Algorithmen sind immer schnell und wahrscheinlich korrekt. Diese Algorithmen liefern nur mit einer gewissen Wahrscheinlichkeit

eine korrekte Antwort. Da diese Algorithmen aber immer schnell sind eignet sich ein solches Verfahren z.B. hervorragend für den hier vorgestellten

randomisierten Primzahltest. Mit Hilfe des kleinen Fermatschen Satzes und dem Test auf nichttriviale Quadratwurzel kann man die Irrtumswahrscheinlichkeit des randomisierten Primzahltests beliebig klein werden lassen. </dc:description>

```
<dc:language rdf:resource="http://www.kbs.uni-hannover.de/Uli/lang.rdf#de"/>
```

```
<dc:subject rdf:resource="http://www.kbs.uni-hannover.de/Uli/ACM_CCS.rdf#I.6.8.6"/>
```

```
<dcq:isPartOf rdf:resource="#Unit2"/>
```

```
</rdf:Description>
```

```
<rdf:Description about="http://ad.informatik.uni-freiburg.de/lehre/ws0102/algorithmentheorie/download/at_lectures/at23.htm">
```

```
<dc:title>Vorlesung Algorithmentheorie WS 2001 (Freiburg) 2.3 Anwendung Randomisierter Primzahltest: Kryptographie (RSA)</dc:title>
```

```
<dc:description>Verschlüsselungssysteme mit geheimen Schlüsseln haben einige gravierende Nachteile. Beispielsweise werden
```

größenordnungsmäßig quadratisch viele Schlüssel benötigt oder die Authentifizierung ist schwierig. Es wird daher auf Verschlüsselungssysteme

mit öffentlichen Schlüsseln eingegangen, die diese Nachteile beseitigen. Es wird erläutert, wie Nachrichten mit öffentlichen Schlüsseln verschlüsselt

werden können und wie man digitale Unterschriften erzeugen kann. Als konkretes Verschlüsselungssystem, das öffentliche Schlüssel verwendet,

wird das bekannte RSA-Verfahren ausführlich vorgestellt. Zur Berechnung des geheimen Schlüssels wird zu einer Zahl die Multiplikative Inverse

einer Zahl benötigt, hierzu wird der erweiterte Euklidische Algorithmus vorgestellt, mit dem man diese Zahl berechnen kann. </dc:description>

```
<dc:language rdf:resource="http://www.kbs.uni-hannover.de/Uli/lang.rdf#de"/>
```

```
<dc:creator>
```

```
<lom:entity>
```

```

    <vCard:FN>Sven Schuierer</vCard:FN>
  </lom:entity>
</dc:creator>
<dc:subject rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#E.3.3"/>
<dcq:isPartOf rdf:resource="#Unit2"/>
</rdf:Description>

```

```

<rdf:Description about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at24
.htm">
<dc:title>Vorlesung Algorithmtheorie WS 2001 (Freiburg) 2.4
Randomisierte Datenstrukturen: Skiplisten, Randomisierte Suchbäume
</dc:title>
<dc:description>Die einfachste Möglichkeit ein Wörterbuch, also eine
Datenstruktur welche die Operationen suchen, einfügen und entfernen
unterstützt,
zu implementieren sind verkettete lineare Listen. Diese haben jedoch
den Nachteil, dass die Wörterbücher-Operationen im schlechtesten Fall
jeweils
linear viele Schritte benötigen. Als Alternative zu balancierten
Suchbäumen, die diese Operationen in  $O(\log n)$  Schritten ausführen
können, wird daher
eine Datenstruktur vorgestellt, die sogenannten Skiplisten, die auf
den Verketteten linearen Listen aufbauen. Zunächst werden perfekte
Skiplisten betrachtet,
die das Suchen in  $O(\log n)$  erlauben. Das Einfügen und Entfernen
benötigt jedoch weiterhin linearen Aufwand. Durch Randomisierung der
Skiplisten
gelingt es jedoch alle drei Wörterbücher-Operationen in erwarteter
Zeit  $O(\log n)$  auszuführen. Im Anschluss daran werden randomisierte
Suchbäume
vorgestellt. Da randomisierte Suchbäume Treaps mit zufälligen
Prioritäten sind, werden daher zunächst die Treaps und die
grundlegenden Operationen
auf ihnen erläutert. Bei den Randomisierten Suchbäumen nutzt man die
Randomisierung um das Degenerieren von natürlichen Suchbäumen zu
linearen
Listen zu verhindern. Auch hier erhält man als Erwartungswert für die
Kosten der Operationen Suchen, Einfügen und Entfernen jeweils  $O(\log n)$ .
</dc:description>
<dc:language rdf:resource="http://www.kbs.uni-
hannover.de/Uli/lang.rdf#de"/>
<dc:subject rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#E.1"/>
<dcq:isPartOf rdf:resource="#Unit2"/>
</rdf:Description>

```

```

<rdf:Description ID="Unit3">
<rdf:type resource="http://telemann.kbs.uni-
hannover.de:3333/olr/olr_v9#Unit"/>
<dc:title>3 : Amortisierte Analyse</dc:title>
<dcq:isPartOf rdf:resource="#Algorithmen"/>
<dcq:hasPart>
  <rdf:Seq>
    <rdf:li rdf:resource="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at31
.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at32
.htm"/>
    <rdf:li rdf:resource="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at33
.htm"/>

```



```
</rdf:Seq>
</dcq:hasPart>
</rdf:Description>

<rdf:Description about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at31
.htm">
<dc:title>Vorlesung Algorithmtheorie WS 2001 (Freiburg) 3.1
Amortisierte Analyse</dc:title>
<dc:description>Bei einer Folge von Operationen, bei der die Laufzeit
einzelner Operationen in einem großen Bereich schwanken kann,
führt die bisher bekannte worst case Analyse zu einer zu groben
Abschätzung, da nicht bei allen Operationen der Folge der
schlechteste
Fall eintreten kann. Es wird daher die amortisierte worst case
Analyse vorgestellt. Bei der Amortisierung wird grundsätzlich
zwischen drei
Methoden unterschieden: Der Aggregat-Methode, der Account-Methode und
der Potentialfunktion-Methode (Bankkonto-Paradigma). Alle
drei Methoden werden anhand eines einfachen Beispiels, des
Dualzählers, ausführlich erklärt. Im Anschluss daran wird die
Potentialfunktion-Methode
noch auf ein komplexeres Problem angewandt, um die amortisierten
Kosten der Operationen Einfügen und Entfernen für eine dynamische
Tabelle zu analysieren.</dc:description>
<dc:language rdf:resource="http://www.kbs.uni-
hannover.de/Uli/lang.rdf#de"/>
<dc:creator>
<lom:entity>
<vCard:FN>Sven Schuierer</vCard:FN>
</lom:entity>
</dc:creator>
<dc:subject rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#F.2.0"/>
<dcq:isPartOf rdf:resource="#Unit3"/>
</rdf:Description>

<rdf:Description about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmtheorie/download/at_lectures/at32
.htm">
<dc:title>Vorlesung Algorithmtheorie WS 2001 (Freiburg) 3.2 Binomial
Queues, Fibonacci Heaps</dc:title>
<dc:description>Vorrangwarteschlangen spielen in vielen Anwendungen
eine wichtige Rolle, wie z.B. beim Kürzeste-Wege Algorithmus von
Dijkstra. Eine sehr effiziente Implementierung von
Vorrangwarteschlangen stellen die Fibonacci-Heaps dar. Zum besseren
Verständnis der Fibonacci-Heaps werden zunächst die Binomial Queues
vorgestellt. Eine Binomial Queue ist eine Vereinigung heapgeordneter
Binomialbäume, wobei die Binomialbäume von verschiedener Ordnung
sind. Bei den Binomial Queues sind alle wesentlichen Operationen in
logarithmischer Zeit ausführbar. Die Initialisierung und Bestimmung
des Minimums benötigt sogar nur konstante Zeit. Bei den Fibonacci-
Heaps muss nicht nach jeder Operation eine so starre Struktur wie bei
den Binomial Queues erzeugt werden. Dadurch ergeben sich bei einigen
Operationen erhebliche Laufzeitvorteile. Es werden ausführlich die
verschiedenen Operationen der Fibonacci-Heaps vorgestellt und mit
Hilfe der amortisierten Analyse deren Laufzeiten analysiert.
Betrachtet man die amortisierten Kosten, so gelingt es mit den
Fibonacci-Heaps alle wichtigen Operationen von Vorrangwarteschlangen
in konstanter Zeit auszuführen, mit Ausnahme der Entferne-
Operationen, die auch hier logarithmische (amortisierte) Zeit
benötigen.
</dc:description>
```

```
<dc:language                                rdf:resource="http://www.kbs.uni-
hannover.de/Uli/lang.rdf#de"/>
<dc:subject                                rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#E.1"/>
<dcq:isPartOf rdf:resource="#Unit3"/>
</rdf:Description>

<rdf:Description                            about="http://ad.informatik.uni-
freiburg.de/lehre/ws0102/algorithmentheorie/download/at_lectures/at33
.htm">
<dc:title>Vorlesung Algorithmenthorie WS 2001 (Freiburg) 3.3 Union-
Find Strukturen </dc:title>
<dc:description>Eine Union-Find Struktur ist eine Datenstruktur zur
Verwaltung disjunkter Mengen. Mit dieser Datenstruktur kann z.B. bei
Graphen
ein Zusammenhangstest durchgeführt werden. Zunächst wird gezeigt wie
man Union-Find Strukturen als verkettete Listen implementieren kann.
Als zweite Möglichkeit der Implementierung wird die Repräsentation
durch einen Wald von Bäumen betrachtet. Es wird gezeigt wie das
Degenerieren zu
linearen Listen verhindert werden kann und wie man für n Union-Find-
Operationen bei einer für die Praxis relevanten Problemgröße eine
annähernd
lineare Laufzeit erzielen kann. </dc:description>
<dc:language                                rdf:resource="http://www.kbs.uni-
hannover.de/Uli/lang.rdf#de"/>
<dc:creator>
<lom:entity>
  <vCard:FN>Sven Schuierer</vCard:FN>
</lom:entity>
</dc:creator>
<dc:subject                                rdf:resource="http://www.kbs.uni-
hannover.de/Uli/ACM_CCS.rdf#E.1"/>
<dcq:isPartOf rdf:resource="#Unit3"/>
</rdf:Description>

</rdf:RDF>
```

Apendix F: IMC „Business Engineer IHK“

Subject Overview

Subject block I

Business Engineer´s Basics

Title	Learning time	Page
Working fields for Business Engineers	ca. 1,5 h	3
Business processes and models	ca. 1,5 h	3
Information systems in enterprises	ca. 1,5 h	4
Information technologies	ca. 4 h	5
Use and development of information systems	ca. 1 h	6
Information system development and procedure	ca. 1,5 h	6
Architectures and frameworks	ca. 1,5 h	7
Business process management	ca. 3 h	7

Subject block II

Business Engineer´s Methods

Title	Learning time	Page
Information modeling	ca. 2 h	9
Data modeling with the ERM – Entity-Relationship-Model	ca. 2 h	9
Forming of relations and normalizing	ca. 2 h	10
Structured Query Language (SQL)	ca. 1,5 h	10
Structured ERM	ca. 1 h	11
Function modeling	ca. 2 h	11
DP-concept functions	ca. 1,5 h	12
Performance modeling	ca. 2 h	13
Organization modeling	ca. 2 h	13
Process-oriented methods	ca. 3 h	14
Knowledge modeling	ca. 3 h	14

Subject block III
Business Engineer's applications

Title	Learning time	Page
Talking and presenting	ca. 2 h	16
Strategic cost management	ca. 2 h	16
Human resource management	ca. 1 h	17
Main processes in industry	ca. 1 h	18
Main processes in public administration	ca. 1 h	19
Multimedia-engineering	ca. 2 h	19
Internet- and Intranet-engineering	ca. 1,5 h	20
Workflow-engineering	ca. 2 h	20
Electronic Commerce	ca. 2 h	21
Knowledge management	ca. 2 h	22
Business intelligence	ca. 2 h	23

Appendix G: RDF Query Exchange Language (QEL)

A new specification is being doing in the context of Edutella project. As it is only a draft it is subject to changes. The current status of the RDF Query Exchange Language (QEL) can be found at <http://edutella.jxta.org/spec/qel.html>.

Here we will only scratch an outline of the current specification:

1. Introduction
 - 1.1 Relationship to other specifications
2. QEL concepts and semantics
 - 2.1 Datalog
 - 2.1.1 Predicates
 - 2.1.2 Facts
 - 2.1.3 Queries
 - 2.1.4 Rules and non-data predicates
 - 2.1.5 Recursion
 - 2.1.6 Datalog Links
 - 2.2 The data space and QEL datalog dialect
 - 2.3 Matching triples
 - 2.4 Built-in Predicates
 - 2.4.1 qel:like
 - 2.4.2 qel:equals
 - 2.4.3 qel:lessThan
 - 2.4.4 qel:greaterThan
 - 2.4.5 qel:nodeType
 - 2.4.6 qel:language
 - 2.4.7 qel:dataType
 - 2.4.8 qel:member
 - 2.5 Negation
 - 2.6 Rules
 - 2.7 Outer Join
 - 2.8 Safe queries
3. Results
 - 3.1 Order of Variables
 - 3.2 Null values
 - 3.3 Results and result sets
4. QEL compliance
 - 4.1 Query Levels
 - 4.1.1 Rule-less Query
 - 4.1.2 Conjunctive Query
 - 4.1.3 Disjunctive Query
 - 4.1.4 Linear Recursive Query

- 4.1.5 General Recursive Query
- 4.2 Unsupported constructs
 - 4.2.1 Does not support rules
 - 4.2.2 Does not support disjunction
 - 4.2.3 Does not support recursion
 - 4.2.4 Does not support general recursion
 - 4.2.5 Does not support a built-in predicate
 - 4.2.6 Does not support outer join literals
- 4.3 QEL implementation capability RDF vocabulary
 - 4.3.1 QEL implementation class
 - 4.3.2 QEL complexity support
 - 4.3.3 Outer Join support
 - 4.3.4 Support for built-in predicates
 - 4.3.5 Support for negation
 - 4.3.6 Example
- 5. Query and result RDF syntax
 - 5.1 Namespaces
 - 5.2 Queries
 - 5.2.1 Variables
 - 5.2.2 Predicates
 - 5.2.3 Query literals
 - 5.2.4 Rules
 - 5.2.5 Queries
 - 5.3 Results
 - 5.4 Self-contained format
- 6. Examples
- 7. References
- Appendix A: Namespaces used in this specification
- Appendix B: QEL RDF schema