Interoperability for Peer-to-Peer Networks: Opening P2P to the rest of the World

Daniel Olmedilla
L3S Research Center and Hanover University
Expo Plaza 1
Hanover, Germany
olmedilla@l3s.de

Matthias Palmér
KMR group at CID/KTH
(Royal Institute of Technology)
Lindstedtsvägen 5
Stockholm, Sweden
matthias@nada.kth.se

ABSTRACT

Due to the information growth, distributed environments are offered as a feasible and scalable solution where Peer-to-Peer networks have become more relevant. They bring many advantages as high flexibility for peers to join or leave the network dynamically, scalability, autonomy and high resilience against peer failures. However, the use of proprietary interfaces within the network and the requirement that peers must implement them to join makes P2P networks unable to interact with other systems and environments, isolating the network as a whole. In this paper, we report on a solution based on a proxy-based architecture and semantic mappings in order to allow the sharing of content between peers within a P2P network with content from other systems outside the network.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; C.2.4 [**Computer Systems Organization**]: Computer-Communication Networks—*Distributed Systems*

General Terms

Distributed Environments, Interoperability

Keywords

Peer-to-Peer, P2P, Proxies, Mapping

1. INTRODUCTION

The World Wide Web has become a common medium for communication among people for private, academic and business affairs. As a consequence, the amount of digital material that is sent along and stored in the network increases rapidly. Obviously, learning is not indifferent to this trend, and the amount of Learning Objects (LO's henceforth) in schools, academy and business continues to grow rapidly. As a consequence of this evolution, the focus shifts to new questions, like for example "Where shall the LO's be stored?", "Who manages them?" or "Are they easily findable?".

WWW2005, May 10–14, 2005, Chiba, Japan.
.

In the past, due to the lack of storage capacity and network bandwidth, especially in most desktop computers, dynamic sharing of information from end user machines was prohibitively costly. As a consequence, networks of computers were mostly reduced to set of powerful connected servers. In this configuration, it is relatively simple to know which servers are available and which information is available where to whom. This is also the typical architecture in business coalitions where several companies share their assets within a network of e.g. partners.

On the other hand, with the boom of Web-based file-sharing services (e.g., Napster, Gnutella, Morpheus), peer-to-peer (P2P for brevity) networks have become more relevant. The advantages of the P2P approach include: high flexibility for peers to join or leave the network dynamically, scalability (recently it was shown that for really large networks, a hybrid solution with super-peers scales better [13]), autonomy as peers do not relinquish control over their resources and high resilience against peer failures. The main disadvantages are that P2P networks require constant management, as peers join and leave continuously (producing an extra load on the network and may slow response times during search) and the use of proprietary interfaces within the network (what usually means that only peers implementing such interfaces can query for or provide content in the network).

Obviously, peers must implement specific P2P network interfaces in order to join them. This means an extra effort for systems willing to connect which already have a different query interface. This barrier makes P2P networks unable to interact with other systems and environments. In this paper, we report on a solution for interoperability in the Edutella P2P network [12] in order to allow the sharing of content from peers within the network with content from other systems and environments outside the network. Our approach is based on a proxy-based architecture as well as on modules that provide semantic mappings capabilities.

The paper is organized as follows: in section 2 we introduce the P2P network we based our work on. The general requirements for interoperability of systems and the assumptions we made for our work are described in section 3. Section 4 describes our proxy based architecture and section 5 introduces the module with semantic mappings. The current configuration of our network is presented in section 6. Finally, section 8 concludes the paper and discusses further work.

2. EDUTELLA

Often, learning object providers do not want to abandon control over their resources to a common server, even among the members of a coalition. The same concern about abandoning control also often applies to individuals, who may not want to give away their content to any centralized repository. Distributed environments have shown to be a feasible solution for interconnection, integration and access to large amounts of information that deal with this issue. P2P networks are an example of the impact this distribution of information might have in the sharing of information. In such networks, peers can offer various services to the user that range from search and delivery to personalization and security services. In addition, they present a solution to the information growth where every learning resource provider offers its information but does not loose the control over it.

The Edutella P2P network [12] was developed with these principles as main design requirements. Edutella [12] is a schema-based P2P network for an open world scenario in which LO's are freely offered (at not charge) and everybody is able to join (no agreement with an existing member of the network is required). It has various service facilities implemented like for example query or publishing/subscription. Schema-based means that peers interchange RDF meta-data (data about data) among each other but not the resources themselves, that is, they interchange information about e.g. title, description, language and authors of a resource. This information can be queried using the QEL query language [14] (a Datalog based query language). Metadata interchange and search services provide the basic infrastructure needed to retrieve information about resources and services.

3. REQUIREMENTS AND ASSUMPTIONS

It is important to note that we consider in this paper only the sharing of metadata about LO's. While this metadata is typically available, the learning object itself might not be. Therefore, we do not deal with negotiations for the actual use of LO's by users here.

Admittedly, providing transparent access to all available repositories would be easy if all players would use the same metadata profile, query language, storage layer and communication protocol. However, this is not going to happen in the very near future due to the lack of a standard and the proprietary solutions adopted by most of them.

In the following, we explain what requirements LO's repositories must satisfy in order to achieve interoperability and which are the assumptions within our network.

Common Communication Protocol and Interface
: Repositories provide different access methods and interfaces, over, among others, Web Services, different Remote Procedure Call methods, HTTP forms or even other appropriate solutions. In order to be able to communicate to each other, it is needed that they agree on a common protocol and a common interface. In this paper, we built on the methods specified in the Simple Query Interface [20] initiative (SQI for brevity), a rapidly maturing standard, using its Web Service binding.

Common Query Language
: At the lower levels of data management, metadata is stored in different kinds of repositories, such as relational databases, RDF repositories, file systems, XML stores, etc. On top of this lower level, repositories expose their content through different search and query languages. Some examples are SQL, XQuery, QEL or CQL. In our system we have several wrappers implemented in order to provide access to the most common repositories (relational databases, RDF repositories, RDF files, etc...). For all them, the wrapper receives a query in QEL and transforms it into the local query language.

Common Metadata Profile
: Although IEEE LOM [1] is becoming a standard for e-learning metadata, many repositories are based on specific profiles that may include extensions and specific value spaces. This means that a mapping needs to be provided [11]. This need even increases when content do not focus only on one domain but extend the content to cover several of them. There are then two possibilities here: either each system maps its schema to a second system schema (in which case we reach semantic interoperability by means of pair of mappings [2]) or a common global schema is provided and both systems must map into that common schema. Section 5 provides a longer explanation and describes a module we developed which allows both approaches.

It is important to notice that although we assume the configuration described above it could be perfectly possible to use a different query language than QEL, a different communication protocol than Web Services and a different interface than SQI though our implementations currently do not support it.

4. PROXYING INTEROPERABILITY

P2P networks are dynamic networks where peers can act as server and client indistinctly and peers might freely join and leave the network over the time. Obviously, peers must implement the specific P2P network interface in order to connect to it. This means an extra effort for systems willing to connect which already have a different query interface. This barrier makes P2P networks unable to interact with other systems and environments.

In order to solve this problem, we based our solution in proxies that are used to connect peers in a P2P network with the "outside" world. This proxies bridge two systems with different capabilities by means of implement the protocol and/or interface supported by each system respectively. This way, a proxy is able to to forward requests and responses from one system to another.

Nowadays, many systems provide their services/resources via Web Services and therefore we implemented proxies able to bridge the proprietary JXTA/Edutella protocol[1] and interface into a Web Service protocol based on the Simple Query Interface.

Taking the P2P network as a reference, there are two different desirable scenarios [18]:

1. An external consumer/client wants to query content in the P2P network. For example, let us suppose that

[1]Here we use the term "proprietary" to emphasize that this protocol is not standard for P2P networks but it does not mean it is not open. In fact, JXTA/Edutella is opensource and anyone can use it easily.

we would like to offer the content of a P2P network on a web site. The first solution would be to make the web server join the P2P network. However, the load of the server would increase considerably and even some problems could arise in case the server wants to provide content from more than one network (it would need to join all of them). A cleaner solution (and the one we follow in this paper) is to forward the query from the Web Site to the P2P network by means of e.g. Web Services and retrieve the answer with the same mechanism.

2. An external provider wants to offer content to the P2P network. We assume that providers that have already implemented a Web Service based interface will not want to spend time and money in developing the proprietary interface of the network. In contrary, they would like to reuse the one they have which would also ease its administration (as only one interface needs to be maintained).

According to these two scenarios, there are two different types of proxies. The former scenario requires the so-called "consumer proxy" and the latter the so-called "provider proxy" (names are assigned according to the role they play). A consumer proxy acts as a mediator between an external client that wants to query the network and the P2P network itself. A provider proxy acts a mediator in order to provide the content of an external provider into the P2P network.

4.1 Consumer Proxy

As described above in scenario 1, in some cases it is needed to be able to query a P2P network without the need of joining it. A consumer proxy is a peer which is part of the P2P network (and therefore it is able to send queries and receive the answers from it) and which is also able to receive requests and send responses using a different protocol and interface. This way, an external client is able to query the P2P network through the proxy.

In our implementation, a consumer proxy mediates between the Edutella/JXTA and Web Service protocols. As depicted in figure 1, it is responsible for

1. Receiving queries from external clients via SQI

2. Forwarding the query to the Edutella network using the JXTA/Edutella interface

3. Collecting the results sent from peers within the network using the JXTA/Edutella interface

4. Forwarding those results to the requester system via SQI

This simple mechanism allows any system to query the content of the Edutella P2P network without needing to implement its specific interface.

In addition, the proxy can return the results to the client application

- Asynchronously. The results are sent to the client as soon as they arrive to the proxy. This is the typical mechanism in distributed environments as not all the results are generated at once but they must be gathered from the different systems in the network.

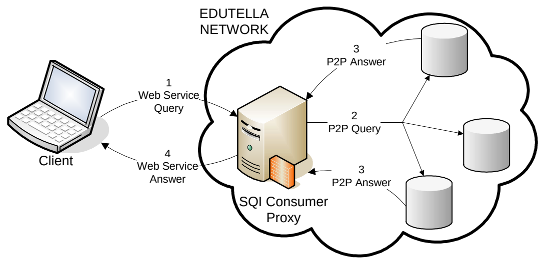

Figure 1: Consumer Proxy

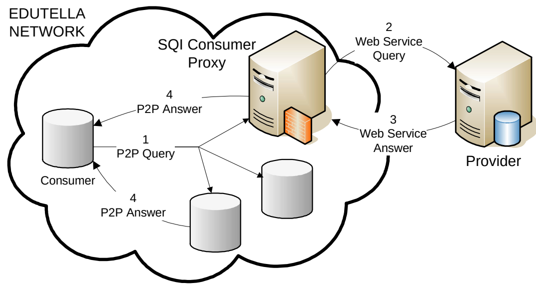

Figure 2: Provider Proxy

- Synchronously. The results are gathered at the proxy and sent together to the client. Although this is not the intuitive way for a distributed environment it could be desirable in some scenarios (e.g., in mobile devices we do not want our device to receive a new message everytime a new result arrives to the proxy but better ask for new results in a proactive manner).

4.2 Provider Proxy

In order to fulfil our scenario 2 a second type of proxy has been developed. This provider proxy is a peer connected to the P2P network which also is able to send requests and receive responses by means of a different protocol and interface. Therefore, it is able to forward queries to external providers and receive their answers providing their content to the network.

As in the the case of consumer proxies, our provider proxy mediates between the Edutella/JXTA and Web Service protocol. As depicted in figure 2, it is responsible for

1. Receiving queries from peers in the network using the JXTA/Edutella interface

2. Forwarding them to the external provider via SQI

3. Receiving the results from the external provider via SQI

4. Sending them back to the peer that sent the query using the JXTA/Edutella interface

Due to the asynchronous nature of a P2P network, it is possible for the provider proxy to receive the results from the external provider in a synchronous (e.g., in case the external provider is a relational database) or asynchronous (e.g., if the external provider is another distributed environment) way.

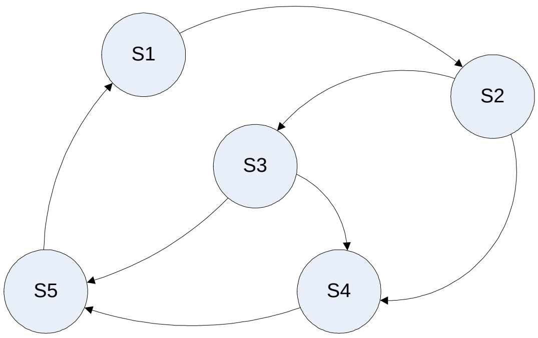

Figure 3: Graph representing 1-to-1 mappings among sources

5. SEMANTIC MAPPINGS

In previous sections, there have been described some of the basis for interoperability, namely common protocol and interfaces (or the use of proxies as presented in section 4) and common query language (or the use of appropriate wrappers). Although these elements ensure that two systems are able to talk to each other it does not guarantee that they will be able to understand each other due to the possible use of different schemas/ontologies.

Nowadays, there is a big effort on standardization of domain ontologies. For example, Dublin Core [4] is intended as standard for cross-domain information resource description and LOM [1] describe attributes required to fully/adequately describe a Learning Object. Unfortunately, still many proprietary schemas are used in each domain (e.g., database schemas within companies). For example, Dublin Core suggests using the attribute "creator" to describe the responsible of making or writting a resource. While many repositories probably follow this suggestion when annotating their resources, others might use e.g., their own attribute "author" instead. In order to bring interoperability among them, data integration in the form of semantic mappings is needed. In this context, a semantic mapping is a transformation from one data model to another data model according to a set of rules (mappings).

In a distributed network we can distinguish among several integration possibilities:

- If no virtual and unified schema is assumed in the network, then systems within the network must provide pairs of mappings between each two systems. Then the distributed network can be seen as a directed graph (as shown in figure 3) in which each arrow represents an available mapping from one node to another. Then, they can be applied transitively in order to infer new mappings which were not explicitly defined. This is specially useful in P2P networks as it is usually not possible to enforce a unique and common schema. Authors in [2] study this approach and provide algorithms to estimate the correctness of the inferred mappings.

- If a virtual and unified schema is assumed, there are two approaches for providing integration between the global schema and local schemas at the sources:
  - Global As View (GAV) [7]. In this approach, the global schema is expressed in terms of the data sources (en example is depicted in figure 4).
  - Local As View (LAV) [23]. In this approach, each source is defined as a view over the global schema. This way, the global schema is specified independently from the sources (an example is depicted in figure 5).

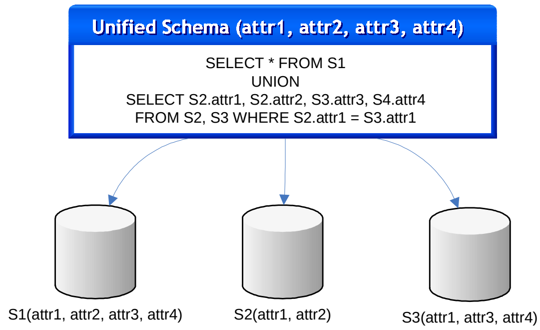

Figure 4: Global As View approach

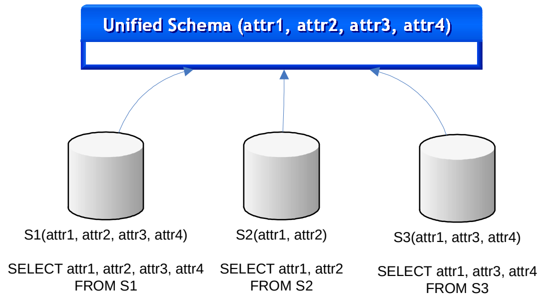

Figure 5: Local As View approach

A discussion of both GAV and LAV is provided in [10] as well as an introduction to "query rewriting" mechanisms. Query rewriting is the process in which a query expressed in the global schema is reformulated into another query according to a set of mappings. This is the mechanism we have used in the mappings module we describe subsequently.

Query Rewriting Module

In order to provide semantic interoperability in our network, we have developed a module which transforms a query q_1 into a query q_2 according to the set of mappings specified. This module is intended to work on pairs of mappings without a unified schema or in Local As View integration approaches.

QEL, the language we use in our network, is based on datalog. In addition to standard datalog constructs, QEL includes some built-in predicates. Taking into account that in our network only metadata (in RDF) is queried and exchanged, the most important one is

qel:s(Subject, Predicate, Object)

which according to the QEL specification [14] "is true if Subject and Predicate are anonymous or non-anonymous RDF resources, and Object is a non-anonymous or anonymous RDF resource or an RDF Literal and the triple Resource

Predicate Object exists in the RDF data". For example, a query like

?- qel:s(X, dc:title, 'Artificial Intelligence').

will return all the resources which title is "Artificial Intelligence". Other useful built-in predicates are $qel : like(X, Y)$ ("used to determine whether an RDF literal or URI contains a string as a substring"), $qel : lessThan(X, Y)$ and $qel : greaterThan(X, Y)$ which are used to compare two RDF literals.

Given this short introduction to the language, let us present the following simple query that we will use for our examples in the rest of the section:

@prefix qel: <http://www.edutella.org/qel#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix lom: <http://ltsc.ieee.org/2002/09/lom-rights#>.
?- qel:s(X, dc:title, Title),
qel:s(X, dc:description, Description),
qel:s(X, dc:creator, Creator),
qel:s(X, lom:cost, Cost),
qel:s(X, dc:subject, Subject).

This query retrieves all the resources with title, description, creator and subject attributes from Dublin Core and the cost from LOM. The first lines of the query with prefix "@" define the namespaces.

Given such a query, we identified the following requirements

- The query specifies a property[2] that does not exist in the source but the source has an equivalent property which could be used instead of. For example, if one data source has its own schema where it uses the property "abstract" instead of the property "description" from the Dublin Core standard.

- The query specifies a property and one value according to a specific taxonomy and the source uses a different taxonomy. For example, if the query searches for resources with dc:subject following the ACM classification [3] and the data source does have dc:subject but it follows the Dutch Basic Classification [5].

- In general, if one of the attributes is not available at the data source, the whole query fails[3]. However, it might happen that although the source does not have explicitly such an attribute, all its resources would share the same value if it existed. For example, assume a repository where all the resources are offered for free. This repository does not have the property "lom:cost" because it is not needed. However, in case one query contains this attribute, the whole query would fail (even if the constraint in the query is "lom:cost = No" which is actually true though it is not annotated). In such a case, it is desirable to assign a default value to all the resources in the data source without having to explicitely annotate all the resources of the repository.

In order to satisfy these requirements we developed a module that performs two types of mappings and one extra transformation: property mapping, property-value mapping and default value transformation (see table 1 for the whole list of mappings and [15] for technical details).

[2]In the paper we will use property and attribute indistinctly

[3]Here we assume that only conjunctives queries are sent. Edutella and QEL support disjunctive queries but we will omit them here because of simplicity

Property Mapping

A property mapping specifies how one property in the query must be reformulated. Its general syntax is

$$(X, p1, Z) \leftarrow (X, p2, Z)$$

When the mapping module receives a query that contains the triple $qel : s(X, p1, Z)$ it rewrites it into $qel : s(X, p2, Z)$. Using our example query and taken into account the requirement in which the source does not contain the property "dc:description" but "own:abstract" (where own stands for their local namespace), it is possible to define the following mapping:

$$(X, dc : description, Z) \leftarrow (X, own : abstract, Z)$$

This mapping is currently a 1-to-1 mapping, that is, there is only one triple at each side of the mapping (separated by the left arrow) but it is also possible to specify 1-to-2, 2-to-1 and 2-to-2 mappings (see table 1). For example, suppose the author in the source is encoded using the property full name from the vcard ontology [19]. In such a case, we need the following mapping

$$(X, dc : creator, Z) \leftarrow (X, dc : creator, Y), (Y, vcard : fn, Z)$$

in order to abstract from the internal representation at the source.

Property-Value Mapping

The mapping described above assumes that one property is completely mapped into another one. However, mapping can be brought to the granularity of values. A property-value mapping applies only when a query contains not only a specific property but also a specific value for that property and then both of them map into other (possibly the same) property and value. Its syntax is

$$(X, p1, v1) \leftarrow (X, p2, v2)$$

For example, assume that our example query uses the ACM classification in the property "dc:subject" and our source does have the property "dc:subject" but annotated with the Dutch Basic Classification taxonomy. We could use several mappings of the form

$$(X, dc : subject, 'Software/Programming\_Languages') \leftarrow$$
$$(X, dc : subject, 'Computer\_Science/Programming\_Languages')$$

to specify how the different values from the ACM taxonomy map into the Dutch Basic Classification.

In the same way as the property mapping, it is possible to extend this 1-to-1 to 2-to-1, 1-to-2 and 2-to-2 mappings.

Default Value

Property and property-value mappings provide rules which define how some triples are reformulated into another triples. The way default values work is a bit different. The properties especified in default values do not exist in the source repository and therefore they must be removed (not just reformulated) in the new query.

Following this approach, when a query is received by our mapping module, if there exists in the query any occurrence

Table 1: **Types of Mappings**

Mapping type	Description
1-to-1 property mapping	$(R, p1, O) \leftarrow (R, p2, O)$
1-to-1 property-value-value mapping	$(R, p1, v1) \leftarrow (R, p2, v2)$
2-to-1 property mapping	$(R, p1, O), (O, p2, L) \leftarrow (R, p3, L)$
2-to-1 property-value mapping	$(R, p1, O), (O, p2, v1) \leftarrow (R, p3, v2)$
1-to-2 property mapping	$(R, p1, L) \leftarrow (R, p2, O), (O, p3, L)$
1-to-2 property-value mapping	$(R, p1, v1) \leftarrow (R, p2, O), (O, p3, v2)$
2-to-2 property mapping	$(R, p1, O), (O, p2, L) \leftarrow (R, p3, O), (O, p4, L)$
2-to-2 property-value mapping	$(R, p1, O), (O, p2, v1) \leftarrow (R, p3, O), (O, p4, v2)$
Default value	$(p \leftarrow v)$

of a property specified in the default values, this ocurrence is temporarily removed. This way, the query is sent to the local repository without that property (otherwise the query would fail) and a resultset is returned. However, this resultset still does not contain the default values that were requested (the properties previously removed) and therefore they must be added. Therefore, default values are added to each of the rows in the resultset returned by the repository.

For example, following with our example query, suppose that our source repository does not have the property "lom:cost" but all the resources in the repository are free of charge. We can then define the following default value

$$(lom : cost \leftarrow' No')$$

This way, any triple in the query referring to the property "lom:cost" would be removed before the query is sent to the repository and added subsequently to the returned resultset together with the default value "No".

6. DRAWING UP THE WHOLE PICTURE

Using the elements described previously in this paper, it has been possible to bring interoperability to Edutella providers (Media Library, Nature and Technology and Confolio System), external consumers (ARIADNE) and external providers (ULI and ARIADNE) within the context of the EU Network of Excellence Prolearn [17]. The picture with the whole architecture is depicted in figure 6 and the following subsections provide a brief description of each of the systems involved.

6.1 Edutella Nodes

In this section we introduce some of the existing peers which are currently connected directly to the Edutella P2P network. All of them are mapped into a common schema (LAV approach). In addition, all the modules used by the peers (including the mapping modules) make use of the highly configurable java library provided in the Edutella project and that it is available from the Edutella CVS (at `http://edutella.jxta.org/`). Some of them, like the mapping modules, are implemented in a way that they can also be used in environments where the Edutella P2P network is not present.

6.1.1 Media Library - Swedish Educational Radio and Television

The media library is a joint project between the KMR group-[9] and the Swedish Educational Radio and Television-[24] - UR in swedish - to provide their educational resources in the form of television programs, radio shows, instructional material, web sites, physical distributions on CDs and videos etc. The educational resources are expressed using the RDF-binding of IEEE/LOM, qualified Dublin Core and some extensions specific to UR. For storage solution we use SCAM-[16] which is a layer on top of Jena2 providing access control on metadata records and a more high-level API. A database provides persistance according to the Jena2 database layout. Queries in the QEL language is directly translated to SQL and passed to the database. We use a library called GETSQL[4] for doing the translation. GETSQL can handle disjunction, rules, most of the constraints, outer join, and also the retrieval service [6] of Edutella.

The peer does two kinds of semantic mappings:

1. simple to qualified Dublin Core mapping. Most queries ask for a title via the predicate dc:title and expects a literal as object. In the Qualified schema there is an intermediate rdf:Alt containing language translations of the title. The mapping is done by rewriting the query to contain an intermediate variable that is not visible in the result.

2. a property-value mapping between two swedish taxonomies, one for the library community and the other for the SLI (Swedish Learning material on the Internet) community.

6.1.2 Nature and Technology - Swedish National Agency for Education

This peer provides educational resources provided by Swedish highschool teachers to be shared. The educational resources are of several kinds, e.g. experiments, articles, projects, tests. The resources are expressed with simple Dublin Core and a few extra properties. The classification is done via the SLI community's taxonomy. The storage solution is SCAM and the GETSQL library is used to translate the QEL queries to SQL here as well. The mappings provided are similar as for the media library with the difference that here we translate from qualified to simple Dublin Core.

6.1.3 Confolio system - portfolios hosted at Royal Institute of Technology

The confolio system is another application built on top of SCAM. It provides users with a directory structure where

[4]GETSQL is expanded to 'Generic Edutella query language Translation to SQL' and is available from Edutella CVS. Its modular design allows the database and database schema to be changed easily.

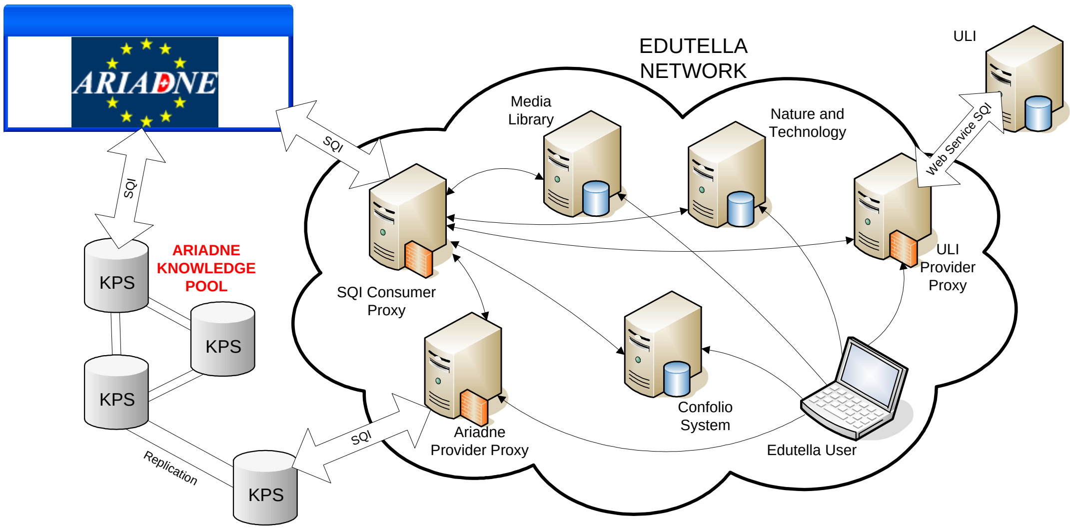

Figure 6: Whole Architecture

they can upload files, store links, or just pure metadata e.g. events, persons, books, concepts. The confolio system allows you to define what kind of metadata that you want to provide for a specific type of resource (multiple types on a resource results in the union of the metadata). A consequence of this flexibility is that there is no specific schema since it depends on what people are doing with the system, on the other hand there is a lot of reuse of Dublin Core, IEEE/LOM etc.

6.2 External Consumers - ARIADNE

In its aim to facilitate both academic education and corporate training, the ARIADNE Foundation supplies its members with tools and methodologies for producing and reusing learning objects. The core of these services is a distributed network of repositories that replicate content and metadata. Doing so, each node contains a copy of all metadata instances. The LO's however can only be replicated to other servers if no download restrictions apply to them. This infrastructure, also known as the Knowledge Pool System, enables the ARIADNE user community to transparently manage learning objects.

The ARIADNE Knowledge Pools offers a client-server approach, where applications can query the ARIADNE knowledge pool through a web services layer. As metadata is replicated in this distributed network, there is no need to federate queries in ARIADNE. However, in order to provide these applications with access to a bigger pool of learning objects, a federated search layer has been built on top of different SQI targets (Edutella, ARIADNE, Merlot, Celebrate) enabling applications to search beyond the borders of the ARIADNE knowledge pool.

Technically, ARIADNE contains metadata information of LO's stored in a relational databases, and the results returned while answering queries is delivered as XML instances following the LOM standard. This information is transformed by means of XSLT stylesheets in order to conver it into RDF.

Currently, ARIADNE is connected to Edutella as external consumer (Edutella content is offered in ARIADNE) and as provider (the content of ARIADNE is offered in Edutella) [21]. As a provider, currently only an ARIADNE KPS node is interconnected and, as a consequence, only ARIADNE material (and not from other systems connect to ARIADNE) can be queried from the Edutella network. Figure 6 shows the current configuration.

6.3 External Providers - ULI

The course repositories ULI [22] (Virtual Computer Center Curriculum in Germany) has been developed under the ULI project (University teaching network for computer science) which tries to establish an exchange of course material, courses and certificates in the area of computer science. Resources include Course-s and Unit-s of study.

Though the courses usually differ in the kind and amount of learning materials they use, their use of learning resources is surprisingly homogeneous. The average course is divided in 6 to 7 units or knowledge modules which themselves can be split into 3 to 7 learning resources. This leads to an average number of about 35 learning resources per course, with a learning resource being the slides of the lecture, a video or any other set of pages dealing with on subject.

Technically, ULI repositories are based on RDF files with Dublin Core and LOM metadata and they are currently accessed by means of the RDQL query language. In addition, mappings and default values have been specified in order to convert to a global schema.

7. RELATED WORK

In [18], the authors describe the two scenarios, consumer proxy and provider proxy, and implemented a translation from the JXTA protocols to Web Services and viceversa. In this paper, we enrich our proxies with the possibility of mapping query languages and schemas. In addition, [18] does not use any specific interface but wrap the java objects while in our approach we are using the SQI standard innitiative.

[8] presents an interesting approach for interoperability of Learning Repositories. Authors briefly present an "ECL Gateway" which is similar to our idea of proxies. They im-

plemented a translation between ECL and a P2P protocol. In our paper we extend this idea separating the two different scenarios, consuming and providing information, and describing in detail how proxies work and how mappings can be performed. In addition, we add our work on default values which, to our knowledge, has not been described yet in any paper.

8. CONCLUSIONS AND FURTHER WORK

In this paper, we showed how by means of proxies and semantic mapping it is possible to connect a P2P network like Edutella with other systems outside the network. These proxies provides the necessary mediation between the different protocols and intefaces and semantic mappings overcome the problem of schema heterogeneity. Both together allow external systems to query and provide content in the network avoiding the isolation of P2P networks from the rest of the world.

In this paper we have focused the interoperability problem on search. However, although this is of course the most important service, there are still some other issues that must be researched. One of the main topics we plan to research on in the future is distributed ranking algorithms. Currently, a lot of research has been done around web ranking (e.g., on the Web) and merging of ranking lists (e.g., on meta-search engines). However, the former assumes that relationships of the form of links exist among resources in different repositories and the latter assumes that there exist overlapping in the content different repositories offer and rank. Unfortunately, this does not apply in a P2P network and the only existing measures are based on trust/reputation of the peers.

In addition, a challenge for the Local As View mappings described in this paper is how they would work in combination with Edutella Retrieval. Edutella Retrieval [6] is a recent addition in Edutella which allows information to be retrieved without requiring it explicitly in the query which would unneccessary eliminate valid matches. Since what is retrieved is not explicitly stated in the query it is far from obvious how to detect which mappings to apply.

Acknowledgments

The authors thank Stefaan Ternier for useful discussions and his work in connecting the ARIADNE system. The research of Olmedilla was partially supported by the IST project ELENA (`http://www.elena-project.org`). The research of Palmer was partially supported by the IST Network of Excellence PROLEARN (`http://www.prolearn-project.org/`).

9. REFERENCES

[1] 1484.12.1 IEEE standard for learning object metadata. june 2002. http://ltsc.ieee.org/wg12.

[2] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. The chatty web: emergent semantics through gossiping. In *International World Wide Web Conferences*, Budapest, Hungary, may 2003.

[3] ACM classification. http://www.acm.org/class/1998/overview.html.

[4] Dublin core metadata innitiative (DCMI). http://dublincore.org/.

[5] Dutch basic classification codes. http://www.kb.nl/vak/basis/bc98-en.html.

[6] ERS: edutella retrieval service. http://edutella.jxta.org/spec/retrieval.html.

[7] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal: Very Large Data Bases*, 10(4):270–294, 2001.

[8] M. Hatala, G. Richards, T. Eap, and J. Willms. The interoperability of learning object repositories and services: Standards, implementations and lessons learned. In *13th World Wide Web Conference (WWW'04)*, New York, USA, May 2004.

[9] KMR: knowledge management research group. http://kmr.nada.kth.se.

[10] M. Lenzerini. Data integration: A theoretical perspective. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 233–246, Wisconsin, USA, jun 2002.

[11] J. Najjar, E. Duval, S. Ternier, and F. Neven. Towards interoperable learning object repositories: The ariadne experience. In *IADIS International Conference WWW/Internet*, Algarve, Portugal, nov 2003.

[12] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A P2P networking infrastructure based on RDF. In *11th International World Wide Web Conference (WWW'02)*, Hawaii, USA, jun 2002.

[13] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Löser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In *12th International World Wide Web Conference (WWW'03)*, Budapest, Hungary, may 2003.

[14] M. Nilsson and W. Siberski. RDF query exchange language (QEL) - concepts, semantics and RDF syntax, http://edutella.jxta.org/spec/qel.html, 2003.

[15] D. Olmedilla. Working with edutella. technical report. http://www.l3s.de/ olmedilla/projects/edutella/edutella.pdf.

[16] M. Palmér, A. Naeve, and F. Paulsson. The scam framework: Helping semantic web applications to store and access metadata. In *ESWS*, pages 167–181, 2004.

[17] PROLEARN: professional learning. http://www.prolearn-project.org/.

[18] C. Qu and W. Nejdl. Interacting edutella/JXTA peer-to-peer network with web services. In *2004 International Symposium on Applications and the Internet (SAINT 2004)*, Tokyo, Japan, jan 2004. IEEE Computer Society Press.

[19] Representing vCard objects in RDF/XML. http://www.w3.org/tr/vcard-rdf.

[20] SQI: simple query interface. http://www.prolearn-project.org/lori/.

[21] S. Ternier, D. Olmedilla, and E. Duval. Peer-to-peer versus federated search: towards more interoperable learning object repositories. In *2005 World Conference on Education, Multimedia, Hypermedia & Telecommunications (ED-MEDIA)*, jul 2005.

[22] http://www.uli-campus.de/index_en.html.

[23] J. D. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239(2):189–210, 2000.

[24] UR: swedish educational radio and television http://www.ur.se.