Peer-to-Peer versus Federated Search: towards more Interoperable Learning Object Repositories

Stefaan Ternier
Katholieke Universiteit Leuven
Leuven, Belgium
stefaan.ternier@cs.kuleuven.ac.be

Daniel Olmedilla
I3S Research Center and Hanover University
Hanover, Germany
olmedilla@i3s.de

Erik Duval
Katholieke Universiteit Leuven
Leuven, Belgium
erik.duval@cs.kuleuven.ac.be

Abstract: This paper reports on our experiences in bridging the world of learning object repositories and peer-to-peer learning networks. More specifically, we have been developing interoperability “bridges” between the ARIADNE Knowledge Pool System, a distributed client-server based learning object repository, and the Eduetella peer-to-peer learning network. In our developments, we rely heavily on the rapidly maturing Simple Query Interface (SQI) standard. Our work not only demonstrates that it is possible to interconnect more centralized repositories and more distributed peer-to-peer approaches. We also clarify how these two approaches are complementary. Unifying the two paradigms for learning object management and access will help to accelerate the evolution towards a critical mass of easily available, relevant learning objects of high quality.

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?”.

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. Searching for information in such a network typically consists of querying each of the systems (known in advance) and of gathering and joining the results received from each of them. This approach is known as Federated Search. The advantages are that search is fast, the more so as this approach also allows specific optimization techniques based on the structure of the network. A drawback is that, when a new entity wants to join the network, it must agree with an existing member of the network on its incorporation into the network.

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 [14]), autonomy as peers do not relinquish control over their resources and high resilience against peer failures. The main disadvantage is that the P2P network requires constant management, as peers join and leave continuously. This produces an extra load on the network and may slow response times during search.

Therefore the answer to the questions stated at the beginning of this section is context dependent. For example, on the one hand, Ariadne [1] is an association that enables share and reuse of LOs through a distributed repository that relies on a client server architecture. On the other hand, Edutella [13] is a schema-based P2P network for an open world scenario in which LO’s are freely offered (at no charge) and everybody is able to join (no agreement with an existing member of the network is required).

In this paper, we build on the Simple Query Interface [7, 17] initiative (SQI for brevity), a rapidly maturing standard for federated search across learning object repositories, to report on our experience in bringing the P2P and client-server architectures together. We demonstrate that they are not only compatible, but also complementary.

The paper is organized as follows: Section 2 explains the basic motivation for this work. Section 3 briefly explains how the Simple Query Interface (SQI) works. Finally, section 4 draws how both the ARIADNE & edutella architectures build on top of SQI.

## 2 Interoperability of LO Repositories

LO’s can be found on the web, as well as in LO repositories. However, as these repositories are often not interconnected, users cannot easily get access to all of the relevant material. Rather, users must know which repositories to search, how to locate them, and then search them individually. This increases user search time and decreases satisfaction. The European IST project ELENA [3] and the European Network of Excellence PROLEARN [6] try to remedy this situation. On a more organizational level, the recently launched GLOBE consortium addresses the same problem: the promise there is to enable federated search across some of the most known LO repositories in the world (ARIADNE, MERLOT, EDNA, EDUSOURCE and NIME) and thus to trigger global interconnection of all such repositories.

In this section, we start with a brief description of our experience in connecting two learning object systems and we generalize from our experience to analyze requirements for interoperability in this context.

### 2.1 Ariadne and Edutella

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.

Often, learning object providers do not want to abandon control over their resources to a common server, even among the members of a coalition. In ARIADNE, this concern is addressed through the distributed nature of the Knowledge Pool System, which enables every organization to set up and maintain its own local server. However, the same concern about abandoning control also often applies to individuals, who may not want to give away their content to any centralized repository. More distributed environments have shown to be a feasible solution for interconnection, integration and access to large amounts of information that deal with this issue. Peer-to-peer 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 [13] was developed with these principles as main design requirements. It is a schema-based P2P network with 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 [15] (a Datalog based query language). Metadata interchange and search services provide the basic infrastructure needed to retrieve information about resources and services.

2.2 Requirements

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.

- **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 (e.g. SQL, XQuery, QEL or CQL).

- **Common Communication Protocol.** Repositories provide different access methods and interfaces, over, among others, Web Services, different Remote Procedure Call methods or HTTP forms.

- **Common Metadata Profile.** Although IEEE LOM [8] 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 [12].

The benefits of the interoperability that these requirements guarantee are twofold:

1. a critical mass of information can be made available as the systems can behave in a virtually unified way;
2. front end applications do not need to be tied to a specific storage layer of a specific repository: a common interface makes it possible to reuse tools (visualisation, query, indexation, LMS ...) over different repositories and an Application Program Interface unites the technical specification (e.g. P2P or Federated Search) from the application layer..

3 Simple Query Interface

For the interconnection of ARIADNE and EDUTELLA, we rely on SQI [7, 17] as a means to interchange queries. SQI is a specification created under auspices of the CEN-ISIS Learning Technologies Workshop. The latter of this paper will clarify how we used this spec in our context.

Currently, 2 specifications are being used SQI: the authentication and session management specification defines different authentication mechanism; the simple query interface specification makes query management possible. As a consequence, a repository can be queried in two steps. First, an application (called source or requester) needs to create a session on a repository (called target or provider). Although the way of creating such a session is not part of the specification, SQI requires a session to be identified by a string of characters. Next, as the session has been established on the target, the source can invoke queries on the target according to two different scenarios (note that the target must provide at least one of both):

1. **Synchronous scenario.** The synchronousQuery method, is invoked with a session identifier and query that are both encoded as a string of characters. As a result, this method will return a configurable number of records that satisfy the query, again encoded as string. Next, the source can invoke the getAdditionalQueryResults method, which returns additional results.

2. **Asynchronous scenario.** After the asynchronousQuery method has been executed, the target is responsible to send the results to source. This method requires the source to host a listener service which is reachable by the target.
In addition to these basic query methods, the specification provides methods to modify parameters that are attached to a session. These methods allow the source to configure, among others, the query language or the number of results returned by the target.

- Whether a query has been issued synchronously or asynchronously, the target will return by default a maximum of 100 results. The setResultsSize method, provides the source with a means to change this amount.
- As some repositories support multiple query languages, the setQueryLanguage method makes switching to another language possible. Note that SQI is agnostic about the query language; neither does it specify the format of the results.

Other methods are for example setResultsFormat, setMaxDuration and setMaxQueryResults.

4 Architecture

In this section we describe the architecture of our approach to making the distributed repository and the P2P network interoperable (see Figure 1). We highlight the three steps that we consider the most important which are from left to right: Ariadne federated search, Ariadne replication mechanisms and Edutella P2P consumer and provider proxies.

![Diagram showing interoperability of federated and P2P networks](image)

**Figure 1. Interoperability of federated and P2P networks**

4.1 Federated Search

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.

Federated search allows clients to search into multiple repositories at once. Compared to searching in caches created by crawlers, federated search takes advantage of the metadata that target repositories already maintain. Also, these searches are conducted on metadata that is always up to date as the repository that owns the documents still manages the metadata.

The ARIADNE federated search layer offers its client applications a SQI interface. In a first step, this layer forwards all queries that are received through this layer to the backhand repositories which also implement an SQI target, supporting the same query language and results format. Next, this layer
collects the results from the different repositories, does some ranking and delivers a default number of results to the client application.

Each query corresponds to a number of threads that are kept open by the federated search layer. Although we can not rely on third party programmers to close sessions properly, we minimized this problem in the ariadne context, where the query tool closes the old session, when a user enters a new query. For contexts where sessions are not closed properly, a timeout mechanism closes the session automatically after 15 minutes.

The possibility to send queries to the SQI API that is offered by this federated search layer, makes developing applications easier. Developers do not need to distribute queries to several repositories and take care of joining the results them selves. Rather can they abstract away from these complexities and use an SQI target that hides this functionality. Analogously, the SQI consumer proxy that will be discussed in section 4.3.1 provides the same kind of abstractions for distributing queries into P2P network.

4.2 Web Services Based Replication

In contrast to federated search, one could opt to distribute the insertions instead of the searches. The ARIADNE Knowledge Pool System is a star shaped network consisting of a number of relational databases, all containing the same metadata.

Each time an item is inserted in one node, the replication framework federates this insert to other ARIADNE nodes. This web services based framework, requires each node to implement 3 methods.

- **Insert**: given an identifier, an xml description of the metadata, an object and credentials this method inserts this new item into the repository. The identifier is necessary to be able to do future updates on the object.
- **Delete**: given an identifier and credentials this methods deletes the metadata and the document that are identified.
- **Update**: requires the parameters as an insert, the method issues a delete first, followed by an insert. One could argue that this method is not strictly necessary as it the same functionality is available using insert and delete. Update however is able to rollback the delete if the insert would fail, which would not be possible if a replication client would use only insert and delete.

This replication framework is similar to the open archives initiative protocol for metadata harvesting (OAI-PMH) [5], a protocol that enables providers to expose their metadata to harvesters. Harvesters are clients that can retrieve records from a data provider. The main difference is that our replication framework pushes data into other repositories while OAI-PMH enables other repositories to pull data from a repository. The advantage of the push approach is that all repositories are at all times up to date, resulting in queries on up to date metadata. Inserting metadata into a repository on the other hand is not a light-weight operation. If these inserts occur very frequently, one might consider pulling instead of pushing.

On an out of the box ARIADNE KPS node, a software layer binds this web service based replication interface to the underlying relational database. In order to provide interoperability to edutella, two interfaces were implemented on top of an ARIADNE RDF repository.

- An interface that implements the edutella provider proxy. This interface enables edutella peers to query ARIADNE metadata in its RDF format. P2P Proxies are covered in the next section.
- An ARIADNE replication interface enables the metadata in the RDF repository to be manipulated. On insertion, an XSLT stylesheet transforms LOM XML instances into the LOM RDF binding which is then stored in the RDF repository. Replication also deletes and updates, this repository manages at all times the same metadata as its other ARIADNE neighbours.

This approach makes ARIADNE perfectly interoperable with Edutella. However, as only an ARIADNE KPS node is currently interconnected, it does not scale up yet to other repositories that ARIADNE is affiliated with. As a consequence, only ARIADNE material can be queried from the Edutella network. The work implemented so far demonstrates the feasibility of our architecture and we plan to achieve full interoperability in the near future.
4.3 Proxies in P2P networks

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 query interface. We built a solution in which proxies are used to connect peers connected to a P2P network with other systems out of the network by means of a common query interface. In our case, we use SQI for the communication between the proxy and the external system.

![Diagram](image)

Figure 2. Proxies in the Edutella P2P network

There are two different kinds of proxies [18] according to their role: consumer proxies and provider proxies. The former acts as a mediator between an external client that wants to query the P2P network and the P2P network itself. The latter acts as a mediator to provide the content of an external provider into the P2P network (Figure 2 depicts the architecture implementation).

4.3.1 Consumer Proxy

A consumer proxy is a peer connected to the P2P network which allows external clients to query the network. It is in charge of receiving queries from external clients via SQI, forward the query to the network using the specific interface for that network, collect the results sends from peers within the network and forward those results to the requester system. This simple mechanism allows any system to query the content of the P2P network without needing to implement its specific interface. This also allows easy integration with web servers which are now able to query the P2P.

4.3.2 Provider Proxy

A provider proxy is a peer connected to the P2P network which allows external providers to provide their content to the network. It is in charge of receiving queries from the network, forward them to the external provider via SQI, receive the results from the external provider and send them back to the consumer that sent the query.

These two kinds of proxies provide with the basis for an easy integration of a P2P network with other kind of systems or networks (including other P2P networks as well).
5 Related Work

Lionshare offers a P2P network that builds on top of the Gnutella P2P network, integrating both centralized and P2P repositories. Compared to Edutella, which builds on top of the JXTA P2P framework, it requires the back-end system not to implement SQI, but one of the following two interfaces:

- The open knowledge initiative (OKI) defines several Open Service Interface Definitions (OSID). The repository OSID specifies the OKI equivalent for SQI, a java interface to be implemented by a repository. This repository OSID lacks random access to the results, as one can only browse through them using an iterator. Also, asynchronous queries are not supported.
- The eduSource Communication Layer (ECL) offers a set of services that implement the IMS DRI specification. Like SQI, this layer supports both synchronous and asynchronous queries.

Apart from the API, these initiatives also differ in query language and encoding of the results. Like ARIADNE, lionshare relies only on IEEE LOM to describe learning resources while edutella is schema independent.

- The digital repository OSID treats queries as not further specified searchCriteria which can be anything that implements the java.io.Serializable interface.
- SQI is also query language independent and only requires queries to be encoded as a string of characters. In the case of Edutella, QEL is used as a query language and RDF as encoding for results. ARIADNE uses a proper XML format to represent queries and returns LOM XML instances.
- The ECL search services closely implements the IMS DRI specification which proposes XQuery as a common query language. A repository that lacks a backhand XML store can still be plugged into an ECL compatible framework, provided that it implements a transformation from predefined XQuery templates to its local query language.
- Lionshare builds on top of the gnutella P2P framework and hence relies on the gnutella XML based format to exchange queries.

6 Conclusions and Further Work

In this paper we presented the main requirements that must be satisfied in order to achieve interoperability among different systems. These requirements include a common query language, a common communication protocol and a common metadata profile. In addition, we showed how to meet those requirements in order to connect systems or networks with different search paradigms. Our experience developing bridges between the ARIADNE and Edutella networks demonstrates the feasibility of making both of them interoperable and shows that both approaches are not only compatible but also complementary. In order to achieve it, we used the SQI specification as basis and implemented different mechanisms like web services based replication and P2P network proxies.

The work presented in this paper proves the feasibility of connecting highly heterogeneous repositories without the need to change the nature of a repository. The unification of different paradigms for learning object management and access provides a big step towards the global sharing and interconnection of LO’s repositories. This helps to accelerate the evolution towards a critical mass of easily available, relevant learning objects of high quality from which users will benefit getting better and more complete results.

In the future, we plan to extend our network to include the other repositories that ARIADNE is affiliated with. This will allow users from the Edutella network to query LO’s from ARIADNE and any of its affiliated repositories. In addition, we still contribute to the SQI standard and its evolution as well as to its possible extensions. Those extensions might be different SQI profiles for communication protocols (e.g. web services) and programming languages (e.g. Java). In addition, we plan to extend the SQI specification with methods that allow not only querying but also data management like inserts, updates and deletions.

Finally, as our main objective is interoperability, we plan to study how to interconnect systems implementing the Lionshare interfaces with systems implementing the SQI interface (including ARIADNE and Edutella). Again, this would provide a new step towards full interoperability among learning object repositories.
7 References