

The Evaluation of Communication Protocol Infrastructures for Collaborative Learning Object Repository Networks

Ty Mey Eap, Marek Hatala, and Dragan Gašević
School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC, Canada, V3T 2W1
{teap, mhatala, dgasevic}@sfu.ca

Abstract

To promote the sharing of learning objects and the creation of collaborative learning object repository networks, diverse e-learning research communities have developed a number of communication protocol infrastructures. The overlapping of the research in this area is beneficial and provides a variety of solutions to the whole e-learning community. However, there are no guidelines for developers to make decisions when choosing a protocol infrastructure for their network. This paper analyzes the requirements of a communication protocol infrastructure for learning object repository networks, evaluates four leading proposed approaches, and produces guidelines and recommendations to help developers choose a suitable protocol infrastructure for their repository network.

1. Introduction

The growing popularity and necessity of online education has researchers anticipating an abundance of learning materials. Without a doubt, making these learning materials available for reuse will revolutionize the learning industry. The abundance of learning resources will allow instructors to choose the most suitable learning materials for their students. Moreover, if the learning materials are granular and self-contained instructional units, then it will be possible to compose training, courses, or more complex instructional units out of them. Anticipating that learning materials will be designed as self-contained instructional units, people have begun to call any learning material a learning object (LO). The potential of learning objects is enormous, but achieving this potential requires progress in several areas of learning technology research.

Sharing information is a winning business model and provides a strategic advantage. The Global Learning Objects Brokered Exchange (GLOBE) initiative (<http://globe.edna.edu.au/globe/go>) is a good example of strategic alliances. ARIADNE, EdNA, LORNET, MERLOT, and National Institute of Multimedia Education (NIME) in Japan have committed to work collaboratively on a shared vision of ubiquitous access to quality educational content. As the use of online learning increases, the Learning Object Repository (LOR) network becomes an attractive solution for providing faster, cheaper, and better learning resources. The benefits of a LOR network can be immediate. The learning communities do not have to wait long for the return on their investments. Although the majority of currently available learning materials are not self-contained instructional units, learning designers and students are still able to use the materials as references or as examples. In some cases, learning designers can integrate, with some modifications, discovered learning materials in their instructional units. Moreover, the LOR network infrastructure stimulates and eases the development of learning technologies.

Since the late nineties, there have been many initiatives that are essential to enable the sharing of LOs. The first initiative was the definition of *learning object metadata (LOM)*, which bridges human knowledge contained in the object into machine knowledge. It enables learning designers to describe a LO, so that machines can interpret its meaning and content. As a result, machines can perform operations such as searching or classifying LOs based on the associated LOM records. The term learning object repository (LOR) refers, primarily, to a LOM management system. This makes LOM interoperability the key component in the LOR network communication. Although standardization is seen by

many not a perfect solution, LOM standards had provided a basic interoperability communication framework. One must command the effort of IEEE LOM to a standard for learning object metadata, which now has become the most used LOM standard. The next emergent technology was the definitions and specifications of *content packaging*. The technology provides definitions and guidelines for learning designers to package learning materials into an instructional unit [1, 4], which tremendously has simplified the management of LOs. Finally, to further promote the development of LO-based learning technology and exchange of LOs, organizations around the world began working on the infrastructures for LOR networks. The purpose of an infrastructure is to allow end-users to access resources on any repository across a LOR network.

This paper discusses the communication protocol infrastructures that bridge individual LORs into a LOR networks. We concentrate on four leading approaches and conduct a comparative analysis in order to gain better understanding of the issues involved and to provide guidelines and recommendations for the development of LOR networks.

2. Background

This section introduces readers to four communication protocol infrastructures discussed in paper.

The eduSource Communication Layer (ECL). Released in June 2003, ECL is an XML-based messaging middleware protocol infrastructure. The development of ECL started with the eduSource Canada project¹ and continued with the Canadian Learning Object Repository NETWORK (LORNET) projects (<http://www.lornet.org>). ECL expands the flexibility of the Web Services (WS) technologies to provide a smooth mapping between ECL messages with respect to LOM query languages and different LOM formats as well as among versions of ECL messages. The mapping strategy allows calling-application as well as LOR developers to use any type of known query languages and LOM formats. This flexibility makes an ECL service generic and configurable. Since the ECL request message is XML-based, the ECL infrastructure can process the message and redirect the message to the appropriate service handler. To implement an ECL service, LOR developers only have to provide a Java-based ECL implementation that handles the service request

message. In addition, ECL provides different types of API utilities such as a LOM query mapping and a query filtering engine to increase the level of interoperability and help developers with their implementations. Furthermore, ECL has developed different strategies that make the ECL protocol highly interoperable, adaptive, and maintainable. The first strategy is the use the middleware and the UDDI registry to coordinate message mapping, software updates, and service discovery. The second strategy was the modularization of the ECL infrastructure design. The middleware and UDDI registry strategy take care of the changes within the ECL infrastructure, while the modularization tolerates the changes of technologies used in the building of the ECL infrastructure.

Search/Retrieval Web Service and Search/Retrieval by URL (SRW/SRU). Released in February 2004, SRU/SRW is a standard search/retrieval protocol based on Z39.50 semantics (<http://www.loc.gov/standards/sru/>). SRU differs from SRW only in the transport protocol: SRU uses a standard HTTP form-based request while SRW uses a SOAP message-based request. Both use Common Query Language (CQL), which is a formal language for representing Web indexes, bibliographic catalogues, and museum collections. SRW/SRU provides protocol definition to allow LOR developers to implement SRW/SRU services, which are accessible by calling applications using SRW/SRU protocol.

Simple Query Interface (SQI). Released in May 2005, SQI is a collaborative effort of the CEN/ISSS (Comité Européen de Normalization/Information Society Standardization System) Learning Technologies Workshop, to achieve interoperability between learning object repositories, and the EU IST-funded ProLearn to coordinate some of the practical experimentations with the SQI specifications [13]. SQI is focusing on providing a common query interface for learning object repository interoperability. It comes with two documents: a specification for building a LOR network (Learning Object Interoperability Framework), and a specification for managing authentication and sessions (Authentication and Session Management). ARIADNE Foundation (<http://www.ariadne-eu.org/>) and EUN Federation of Internet Resources for Education (FIRE) (<http://insight.eun.org/ww/en/pub/insight/interoperability/monthlyinsight/fire.htm>) are two main SQI promoters and have implemented SQI on two separate platforms: ARIADNE implemented SQI on Web service technology while FIRE implemented SQI on Java Message Service.

¹ An 18-month long Canadian LOR network initiative that started in 2002

Open Source Interface Definitions (OSID). Rather than a protocol, OSID is a collection of interface function definitions that enable an independent implementation of calling applications and service providers. OSID provides a wide range of definitions for common functions in the learning environments such as the assessment, grading, filling, authentication, authorization, logging, etc. OSID has been proposed by the Open Knowledge Initiative (OKI) to promote specifications that describe how the components of a software environment communicate with each other and with other enterprise systems (<http://www.okiproject.org/>). OKI believes that the specifications enable sustainable interoperability and integration by defining standards for Service Oriented Architecture (SOA). The first implementation of OSID, developed by Tufts Visual Understanding Environment, was released in March 2004 (<http://vue.tccs.tufts.edu/about/index.cfm>). However, this paper analyzes OKI/OSID based on the ECL/OSID implementation, which was a part of the LionShare project (<http://lionshare.its.psu.edu/>).

The four approaches presented above cover a wide range of views on LOR networks and are designed, more or less, for heterogeneous and collaborative repository network, which is a criterion for this paper's comparative analysis. Each approach builds the solution based on their views of the LOR network. Hence, each has advantages and disadvantages, depending on the type of the LOR network.

3. Characteristics of Learning Object Repository Infrastructures

There are different views on characteristics of a LOR network. Each view influences the approach and the design of the communication infrastructure. This section examines each approach's views of the LOR network.

ECL presumes that e-learning technology is relatively new and needs rooms to grow and mature. This view requires a flexible and adaptive protocol infrastructure that can be expanded to support the technological evolution. Furthermore, to encourage repositories to share their resources, ECL provides a flexible single-sign-on security infrastructure based on trust. It enables repositories to share learning objects at different levels of security. As a result, ECL provides a complete communication protocol infrastructure that is extendable and supports a variety of services. ECL is designed based on four principles: abstraction, modularity, maintainability, and enable-and-not-require. The **abstraction principle** targets abstraction

of complexity of new technologies and protocol standards by providing simple interface functions or APIs to ease the integration with the ECL infrastructure. The **modularity principle** enforces the grouping of similar technology components in order to reduce dependency on the technologies to permit the substitution of the technologies without affecting the whole ECL infrastructure. The **maintainability principle** forces the development of strategies that makes the infrastructure adaptive to change. For example, a calling application using an older ECL version should be able to invoke the ECL service implemented on a new ECL infrastructure. Similarly, the protocol infrastructure should be able to assimilate new protocol standards and be adaptive to the changes of existing standards. Finally, the **enable-and-not-require principle** states that ECL should not impose any requirement on developers using the ECL. This principle requires the use of a middleware that can perform self-configuration and self-installation of the ECL infrastructure.

SRW/SRU focuses on public archives such as libraries. Large institutions are stable and have well-established protocols for metadata definitions and classifications of the learning materials. The materials are well defined and self-contained (i.e., books, journals, proceedings, articles, etc.). A simple search-and-retrieve protocol as such as SRW/SRU is sufficient to make the materials available to the public.

Similar to ECL, SQI targets heterogeneous and collaborative repository network. The SQI protocol is designed to be neutral in terms of LOM formats and query languages allowing a repository using SQI to use any query language and LOM format. In addition, SQI supports synchronous and asynchronous queries as well as session management and stateful/stateless communication to accommodate different types of repositories. The design of SQI follows **command-query-separation** and **simple command set and extensibility** principles, which make SQI interfaces clearer, more understandable, and extendable. The specifications for LOR interoperability wrap SQI in a framework and provide mapping for LOM and LOM queries. However, the two implementations (ARIADNE/SQI and FIRE/SQI) reviewed by this paper support neither a LOM query mapping nor a LOM mapping. For security, SQI has a simple authentication specification that is tied with the session management. In order to establish a session with a source repository, a target (calling application) must supply identification (username and password) of the source repository. Although the view on the LOR network is similar to ECL, the expectation of SQI on the collaboration among developers is much higher.

ECL anticipates some difficulties and provides framework that helps organize the collaboration among developers. For example, an ECL client can discover and access any repository on the ECL network. The security constraint should be resolved by the ECL network federation. Meanwhile, SQI does not try to resolve ambiguity. A target (client) application developer must negotiate with every repository to gain access the repository's resources. The developer needs to know each repository's query language, LOM format, communication type (i.e. synchronous or asynchronous), and security type.

The OSID is a set of Java interface functions for which OKI believes that they would provide general software contracts between service consumers and service providers. This should enable applications to be constructed independently of any particular service environment, and ease integration. OKI claims that OSID enables choices of end-user tools by providing plug-in interoperability through software contracts that are compatible with most of other technologies and specifications, such SOAP and WSDL. OSID guarantees a stable interface of each OSID implementation for a client application, but the idea only works if the end-user tools use a limited number of plug-in implementations of OSID. It is not ideal if each repository has to provide a plug-in OSID implementation. The discrepancies among OSID implementations can be overwhelming for application developers to handle.

4. Analysis and Evaluation

The goal of the analysis is to see how well each protocol infrastructure responds to the required criteria for an evolving and collaborative LOR network. This paper conducts a comparative analysis in three areas: *architecture*, *integration*, and *interoperability*. The structure and technologies used in the architecture as well as the design decisions have an impact on the infrastructure flexibility. *The architecture analysis* aims to reveal characteristics that may or may not be desirable for a collaborative LOR network. Next, to incorporate a protocol infrastructure service, developers must understand the service syntactically and semantically. *The integration analysis* approximates the complexity of the infrastructure integration based on the number of methods and parameters of the services as well as the protocol structure. Finally, *the interoperability analysis* concentrates on the amount of work that is required to make a protocol infrastructure interoperate with another protocol.

4.1 The Challenges

The usage of multiple LOM formats and LOM query languages are the two major challenges in providing interoperability for LOR networks. This section gives some insights and discusses solutions to these challenges

4.1.1 Metadata. Metadata provides a structure and encoded information, typically in an XML format, about a learning object, which enables a machine to classify, sort, and search a collection of metadata. The management of learning objects is achieved via LOM records. Currently, there are multiple LOM standards and application profiles (i.e. Dublin Core (DC), CanCore, and IEEE), which create a challenge for communication protocol infrastructures to provide the interoperability for repositories using different LOM standards. Mapping between two LOM standards can result in the loss of important information. For a mapping to be successful, there must be a one-to-many or one-to-one condition. Currently, mapping from DC to IEEE LOM does not lose any information, but mapping from IEEE to DC loses a considerable amount of data. Many IEEE LOM elements are undefined in DC. Some may argue that the loss of these data is not significant since DC users do not use the additional data in the IEEE LOM. This is true in most cases, but in some specific cases the loss is significant (e.g., the loss of IEEE LOM identifier references). The mapping guidelines advise mapping an IEEE identifier and IEEE location to a DC identifier. This is a many-to-one mapping, which is impossible to map without a further processing. Another problem is the catalogue entry in the IEEE LOM identifier. IEEE LOM intends to provide flexibility by associating catalogue and identifier for multiple identifiers. However, this flexibility creates a problem in LOM mapping. An IEEE LOM repository may use a generic URL for location, and to request a LO. The repository requires the use of both identifier and location. Hence, the LOM mapping needs additional information. To resolve such ambiguity two systems must agree on the handling of identifiers (i.e., combining location with identifier).

4.1.2 LOM Query Language. LOM query language is an area that needs more research. One of the problems is the incomplete specifications and requirements of a LOM query. However, this cannot be completed until the e-learning technology becomes more mature. The second problem is the complex structure of the LOM format, especially IEEE LOM. Supporting multiple languages, identifiers, taxonomies, etc. accommodate a

wide audience, but this flexibility requires a complex query language. Early in the development of the LOR network research, many researchers thought that XQuery (<http://www.w3.org/TR/xquery/>) was a good candidate for a LOM query language. It made good sense since LOM is represented in the XML format. As a result, eduSource adopted XQuery as the query language for ECL [3]. However, XQuery never caught on. Furthermore, LOM query must deal with different syntaxes of text search. In addition to SQL wild card query, there are Google, Lucene, and Common Query Language (CQL) text searches. To reduce the complexity of LOM query, ECL divides LOM query into two parts: text search and domain specific query (or advanced search) and uses a domain specific query filter engine to compensate for the lack of a domain specific query support in some repositories in order to provide a uniform LOM query throughout the LOR network.

4.1.2.1 Google-style Text Search Syntax. Google has gained tremendous popularity with its simple, tolerant text search. The Google search query is a combination of keywords called term. The order of keywords is a search criterion as well and can influence the search results. By default, Google search ignores stop words such as “on” or “of” and considers all non-alphanumeric characters with the exception of some special characters (e.g., double quote mark ("), plus sign (+), minus sign or hyphen (-), and ampersand (&)) as white spaces. Google only considers the “OR” operator, while the “AND” operator is implicit. In addition, Google offers simple rules for additional filters such as filtering language with “<fr>” or search all text with “(all)” (<http://www.google.com/apis/reference.html>).

4.1.2.2 Lucene Text Search Syntax. Some repositories such as SMETE and ECL-POOL use the Lucene text search engine for the text search (<http://lucene.apache.org/java/docs/queryparsersyntax.html>). Lucene is open source software, and freely available for download, which can be a factor in the development of a LOR network technology. In addition, Lucene can perform a text search on tabular data. The syntax consists of terms, fields, and operators. The terms are regular expression (e.g., “The Right Way”, “Goo~”, “G*gle”, or “Goo?le”). The fields are column names (e.g., title, description, etc.) Finally, the operators are “OR” or “AND”. An example of Lucene query can be following: title: "The Right Way" AND text: go. Undoubtedly, Lucene can be used to store metadata’s text elements, and be used as a primary basic LOM query.

4.1.2.3 Common Query Language (CQL) Text Search Syntax. CQL has a very strong support from repositories such as library and government document archives. As mentioned earlier, it is a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogues, and museum collection information. The Library of Congress is currently in charge of maintaining the CQL standard, which explains the overwhelming support of CQL from librarians. CQL is the core query protocol of SRW/SRU. The query is a combination of words and operators. For example, the following query “title all ‘complete dinosaur’” means retrieve words in title contained ‘complete’ and ‘dinosaur’.

4.2. Limitations

This paper analyzes SQI based on implementations of SQI on ARIADNE and FIRE, which do not implement all the specifications found in the “A Simple Query Interface for Interoperable Learning Repositories” document [13]. The framework described in the document is comparable and similar to the ECL framework. To achieve such interoperability, FIRE must add additional utility libraries for query and LOM mappings, and ARIADNE must be implemented on a middleware. ARIADNE should not expect to achieve the expected interoperability by simply providing a WSDL file. Similarly, it is difficult to analyze OKI/OISD, which is a definition of interface functions. The implementation experience of ECL/OISD had shown that OKI assumptions do not automatically transcend reality. Developers must deal with many issues concerning OSID, some issues are impossible to resolve without the extensions of OSID. For example, the implementation ECL/OISD required the extension of OSID type definitions for IEEE LOM. Finally, SRW/SRU targets a specific type of repositories and only supports a single query language. Hence, SRW/SRU cannot support many criteria required for heterogeneous LOR networks.

4.3. Architectural Comparative Analysis

Over the past decade, there have been a large number of proposed architectural analysis methods, and each focuses on different aspects of the architecture [6]. The method depends on the criteria chosen to conduct the analysis. In other words, a software architecture analysis is an ad hoc analysis. For example, Lindvall et al. proposed an evaluation based on the analysis of maintainability. The method is the restructuring of the architecture by adding new

components and analyzing the discrepancies between the new and old architectures [9].

This paper uses a method similar to the one proposed by Kontio et al. in evaluating reusable software components [7]. The method consists of defining factors, goals, and criteria for evaluating a reusable component. Of course, factors, goals, and criteria in this analysis are based on the need of the collaborative and evolving environment of the LOR network. As Kazman et al. have indicated, the intent of the criteria is to determine how well the architecture satisfies a set of business goals [6]. The criteria chosen for this analysis are based on the requirements for collaborative and evolving LOR networks.

1. *Multi-Query Languages* – the support for multiple query languages
2. *Common Query Language* – the support for a common query language².
3. *Query Mapping* – the indication of the strategy for query mapping.
4. *LOM Mapping* – the indication of the strategy for LOM mapping.
5. *Version Compatibility* – the ability to maintain compatibility among versions
6. *Multi-purpose Framework* – the support for other types of communications
7. *Platform Independence* – the capability to work on different platforms
8. *Discovery Service* – the indication of the strategy to dynamically form a LOR network
9. *Asynchronous* – the support for asynchronous messaging
10. *Interoperability* – the potential to interoperate with other protocol infrastructures.

4.3.1. Architecture Analysis Summary. Table 1 reflects findings based on the selected criteria.

Without the support of a middleware and utility libraries, developers using SQI and SRW/SRU must deal with every issue of the interoperability. By allowing repositories to use any type of query language, SQI transfers the problem to calling application developers. Both, SQI and SRW/SRU, let repository developers decide whether or not they want to support LOM mapping or version compatibility. SRW/SRU repository developers must provide XSLT file to enable the support for a LOM mapping. As a result, calling application developers must integrate each repository individually; they must know the

² While multiple query languages support allows LOR developers to use their favorite query languages, calling application developers should only have to deal with a single query language

repository query language and LOM format. Finally, to guarantee a stable interface, OKI/OSID shifts all responsibilities to the OSID implementation developers.

Table 1. The comparison of infrastructures: Support of architectural features

(√ fully, ? maybe, - not supported or not applicable)

Description	ECL	SQI	OKI	SR(W/U)
Multi-Query Languages	√	√	-	-
Common Query	√	-	√	√
Query Mapping	√	-	-	-
LOM Mapping	√	?	-	?
Version Compatibility	√	?	-	?
Multi-purpose Framework	√	-	-	-
Platform Independent	√	√	-	√
Discovery Service	√	?	-	?
Asynchronous	√	√	-	-
Interoperability	√	√	√	√

4.4. Analysis of Integration Requirements

What is the level of difficulty for developers to integrate, adapt, update, and maintain a protocol service? This paper uses ECL as a base comparison for integration analysis and selects to analyze the implementation of SQI on ARIADNE, which gives better advantage for SQI due to platform dependence of FIRE. To access an ECL LOR network, calling application developers only have to integrate the ECL client API. Developers need to know how to query the ECL UDDI registry, build the query, and process the results. Let us assume that developers know how to process XML documents such as building search requests and processing results. This analysis uses the same method as in the architecture analysis, and assumes that each LOR has a Web server, and the maintenance of a Web application or a Web service is easy.

1. *Overall Integration* considers the overall difficulty for a service provider as well as for a calling application to integrate the evaluating protocol infrastructure.
2. *Protocol update* focuses on the requirements for updating the implementation of the evaluating protocol.
3. *Adaptation* refers to the effort of an existing system using another protocol standard to adapt and use the evaluating protocol.
4. *Maintenance* considers the need to maintain the evaluating protocol service.

4.4.1. Integration Analysis Summary. As shown in Table 2, the levels of difficulties of the ECL integration for all the above criteria are considered to

be easy. ECL is a complete protocol infrastructure that is able to do self-diagnostic and auto configuration. It is designed to be backward compatible. On the other hand, the overall integration and adaptation of ARIADNE/SQI and SRW/SRU can be considered as moderate since both are Web services based. The update of the protocol requires the same effort as a new integration while as it has been mentioned earlier, the maintenance of a Web service is considered to be easy. The integration and update of OKI/OSID are difficult. Developers are facing many obstacles when implementing OKI/OSID. The difficulty increases when each repository has to implement an OKI/OSID plug-in. On the other hand, any existing repository should be able to implement the OSID, and any Java calling application can use an OSID plug-in.

Table 2. The comparison of infrastructures by the integration criteria

(Easy, moderate (Mod), difficult (Diff))

Criteria	ECL	SQI	OKI	SR(W/U)
Overall Integration	Easy	Mod	Diff	Mod
Protocol Update	Easy	Mod	Diff	Mod
Adaptation	Easy	Mod	Easy	Mod
Maintenance	Easy	Easy	Easy	Easy

4.5. Interoperability Analysis

According to the IEEE Standard Computer Dictionary, interoperability is the ability of two systems or components to exchange information and use the exchanged information. Hence, the key to interoperability is the ability of the protocol infrastructure to map from and to a foreign protocol. The problem with mapping is the loss of information. Mapping IEEE LOM to DC LOM can lose more than 30 percent of the data. On the other hand, when mapping from DC LOM to IEEE LOM, the loss of data is practically zero. Based on this observation, an estimation of interoperability between two protocols is possible.

When estimating interoperability, other factors come into consideration. One is significant and insignificant loss of data. The loss of data when mapping IEEE LOM to DC may be insignificant because DC users do not use extra IEEE LOM data. However, in some cases, when mapping IEEE LOM and DC, there is some significant loss of data. For example, IEEE LOM has two sources (identifier and location) to map to DC identifier. Normally, the IEEE LOM location is mapped to the DC identifier. If a repository needs an IEEE LOM identifier to retrieve a LO, DC users do not have the IEEE LOM identifier to

access the LO. Another problem is the difference in the data format. An IEEE identifier has two values: catalogue and entry. In this case, the interoperability requires an agreement between DC users (calling application) and IEEE repository (service provider) to create a new format for the identifier (e.g., a composite identifier using the following form: “catalog:entry”). This example illustrates the issue of mapping and interoperability. Despite the effort, avoiding the loss of significant data can be difficult in some contexts. Similarly, there is no loss of data when mapping from DC LOM to IEEE LOM, but this does not mean 100 percent interoperability. If an application was using IEEE LOM identifiers for some purposes, the mapped IEEE LOM from DC would not have identifiers. How would the application deal with the missing identifiers?

The following provides a brief analysis of interoperability (see Table 3). The analysis considers the interoperability between two protocols in one direction at a time and only considers Web-based protocols such as ECL, SQI/ARIADNE (version 1.0), and SRU/SRW (version 1.1).

Table 3. The evaluation of infrastructures based on the interoperability criteria

(Text Search Mapping (TSM), Advanced or Domain Specific Query Mapping (DSQM))

Protocol	Query Interoperability requirements	Metadata Interoperability requirements
ECL → SQI	Minor work on (TSM)	No work
ECL → SRW ECL → SRU	Minor work on (TSM)	Minor work on LOM Mapping
SQI → ECL	Minor work on (TSM)	No work
SQI → SRW SQI → SRU	Minor work on (TSM) Major work on (DSQM)	Minor work on LOM Mapping
SRW → ECL SRU → ECL	Minor work on (TSM)	No work
SRW → SQI SRU → SQI	Minor work on (TSM)	Some work on LOM Mapping

4.5.1. Integration Analysis Summary. With support of middleware and utility libraries, ECL is able to improve the interoperability with other protocols. For example, ECL can apply query filter on the result from other protocols, which improves the interoperability of query mapping. Similarly, other protocols can request the results in any known LOM format. In contrast, SQI and SRW/SRU need to deal with results when other protocols return a different type of LOM formats. In addition, SQI must deal with the advanced or domain

specific query when other protocols only support text search.

5. Discussion

By coupling with a middleware and a UDDI registry, ECL takes full advantage of the XML-based messaging to provide an integrated infrastructure that is adaptive, extendable, and maintainable. SQI and SRU/SRW follow a similar footsteps as ECL. Both use XML-based messaging protocols. The only difference from the ECL is the way in which they pass the message. ECL passes the message as an XML string over a predefined RPC WS that is deployable and accessible using a SOAP utility without any configuration. Meanwhile, SQI/ARIADNE uses document style WS and SRU/SRW merely sends the message directly over HTTP. OKI/OSID mostly concentrates on providing a stable set of interface functions for the calling application developers. In a developing and evolving environment, OKI/OSID faces many unresolved issues, which make OKI/OSID a difficult infrastructure to use.

Based on the criteria required for an evolving collaborative environment, ECL clearly comes out ahead of all other protocols. It equips developers with utilities and strategies and abstracts the complexity of the technologies and the protocol standards to ease the integration of the ECL into the calling application as well as into the service provider. The protocol is generic, and is capable of supporting other types of communications. In addition, ECL has developed strategies to interoperate with other protocols. The so called gateway strategy of ECL enables ECL clients to invoke services such as SQI/ARIADNE or SRU/SRW services as if the services are ECL services. In comparison, SQI/ARIADNE or SRU/SRW calling application developers are responsible for integrating each service individually.

There is a stereotype that favors the 'simple' WS technologies against the middleware. Some researchers believe that WS are stable and reliable, and everybody knows how to implement and invoke a Web service from a WSDL description. However, there are some complications when implementing a WS from a WSDL file. WSDL is a binding contract that a service provider gives to its calling application developers, and is designed for calling applications to invoke the service. Therefore, the service must be stable and well defined. SRU/SRW has dropped the conventional WS standards in favor of the XML-based messaging over HTTP. This makes the SRU/SRW protocol more attractive, easier to integrate, and expandable. When

communication infrastructure developers want to provide better assistance to LOR developers, they must consider middleware technology or some kinds of utility libraries.

Apart from providing flexibility and easy integration, the XML-based messaging middleware technology enables ECL to develop maintainability and message mapping strategies for interoperability within ECL implementations (i.e. vertical compatibility), as well as with other protocol infrastructures. Combined with the UDDI registry, the ECL infrastructure is able to feed the information on new software and protocol standards to the calling applications using the ECL. This characteristic is different from the software updates where software distributors have full control over the update. The calling application is attached to the ECL infrastructure, and its implementation is based on the API provided by ECL. The integrated application knows how to build a particular format of input request to invoke an ECL service, and expects a particular type of response. Hence, updating a middleware implementation is more complex. The update involves message mapping. An older ECL implementation must be able to map the old ECL message to the new ECL message used by a service provider. This mapping strategy becomes handy when trying to interoperate the ECL with other protocol infrastructures.

However, readers must be aware of the fact that there are other factors that play an important role in the success of an infrastructure protocol. To be successful, a protocol infrastructure must provide support to the developers and continue to develop and respond to the need of the LOR networks. ECL is a research project and is not able to provide such support. Several eduSource partners lost their funding and ceased to provide ECL services. In contrast, SQI has very strong support from the European Commission. SRU/SRW has the backing of the Library of Congress. EdNA, in which the protocol infrastructure is proprietary, has very strong support from Australia educational institutions and government. Similarly, SMETE has the backing of the National Science Foundation. With sustainable funding, the later protocol infrastructures have better chance for success.

6. Guidelines and Recommendations

Based on the previous arguments, ECL is an infrastructure for an evolving collaborative LOR network for any type of LORs (i.e., small, medium, and large). ECL enables the sharing of LOs while assisting in the development of learning technologies.

The ECL infrastructure is generic, can accommodate other types of services, and equips with a UDDI registry for service discovery and a multi-level security infrastructure.

SQI/FIRE is a good candidate for a peer-to-peer type LOR network. It does not require a dedicated server for a LOR, which can be located behind a firewall. LORs (target) and calling applications (source) participate equally in the network and must maintain an open connection to a JMS provider. Hence, the protocol is suitable for small LORs that want to participate in a peer-to-peer network.

For a public or semi-public LOR network that consists of independent LORs where LOR developers only want to provide a common search service to access LOR resources, the suitable protocols are SRU/SRW or SQI-WS of the ARIADNE type. SRU/SRW protocol is easy to implement and is a better choice if CQL search is sufficient for all LORs in the network, SQI-WS may be better suited for a LOR network that needs an advanced search support.

An ideal network for OKI/OSID is a homogenous LOR network. Obviously, calling applications should have only a single OKI/OSID implementation to deal with. The goal and purpose of the OSID is to provide simple and stable interface functions for calling applications. Therefore, it does not make sense to present multiple OKI/OSID implementations to the calling application developers. Consequently, OKI/OSID needs an underlying protocol infrastructure that can unify heterogeneous LORs or a homogenous LOR network.

6.1. Implementation Guidelines

Separating a text query from a domain-specific query – the strategy reduces the complexity of the metadata query and allows communication protocol developers to resolve the problem with a text query before focusing on the domain specific query. For example, the text query must deal with two issues: the language and text search syntax. The language used on the application operating system can be the default language for the text query, and the syntax can be the Google, Lucene, or CQL text search syntax.

Providing a domain-specific query filter is obviously a good idea. LORs cannot adapt quickly enough to support new requirements of a domain-specific query. A protocol infrastructure should be able to apply the domain-specific query filter on the client side on the query results before presenting them to the end-users. This would unify the search on a network.

7. Other Protocol Infrastructures

The first related work that can create an impact on e-learning management systems is the Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA), an open standards-based model. The CORDRA initiative has its roots in the US Department of Defense's Advanced Distributed Learning (ADL) initiative (<http://www.adlnet.org/>), well known for the Sharable Content Object Reference Model (SCORM) [8]. CORDRA aims to identify and specify (not develop) appropriate technologies and existing interoperability standards that can be combined into a reference model that will enable learning content to be found, retrieved, and re-used (<http://cordra.net/>). The model assumes a set of locally managed repositories, augmented by an object identifier infrastructure, common services and applications, and three kinds of repositories: a master catalogue that registers the content held in the federation, a registry of participating repositories, and a system registry [8]. According to Kraan's report at the First International CORDRA™ Workshop [8], the stakeholders in e-learning management had raised numerous issues on CORDRA approaches. Consequently, the workshop organizer, Dan Rehak, emphasized in his summary that CORDRA would not be finished for a while yet and the workshop was only the first step. Finally, Kraan concluded in his report that for CORDRA to be successful, CORDRA must determine the minimal components that are technically feasible.

The next related work is GLOBUS, a community of users and developers who collaborate on the use and development of open source software as well as associated documentation for distributed computing and resource federation [5]. GLOBUS released GLOBUS Toolkit (GT) in the late 1990s, a set of libraries and programs addressing common problems that occur when building distributed system services and applications. Once installed, a GT becomes a grid cell that can be configured to communicate with other GT grids that are part of a GLOBUS grid ecosystem. GT is another example of a middleware that uses WS technologies for its core components. The first two steps for implementing a GT Web service are similar to the implementation of a Web service from scratch without using the SOAP utility: defining the service's interface by writing a WSDL file for the service and implementing the service in Java. Then, the developers must prepare the service by writing the configuration parameters into a configuration file and compiling everything into Grid Archive or GAR file. Finally,

developers can use GT tool to deploy the service into the grid ecosystem.

There was some interest in using GLOBUS framework for an e-learning management system. Reklaitis et al. proposed in 2003, the mapping of e-learning components onto a set of standard GLOBUS services to accommodate the distribution of e-learning applications on the grid and to view the applications as typical grid services [10, 11]. However, Reklaitis admitted that he and his team needed to solve many questions before they can realize an e-learning implementation according to the grid service specifications. In the last two years, the interest in the GLOBUS for e-learning management system has disappeared. GLOBUS is a closed system. Each e-learning application must be implemented as a GLOBUS service. GLOBUS is suitable for managing resources of an organization, but it is unreasonable to impose the installation and the maintenance of a GLOBUS system on an existing repository.

Finally, LionShare and SPLASH deserve some considerations as related works. Both are distributed peer-to-peer networks of object repositories, and both had integrated the ECL infrastructure to access the ECL network. LionShare is a secure peer-to-peer network sitting on top of the Limewire infrastructure, which is a peer-to-peer system based on the Gnutella protocol (<http://www.limewire.com>). SPLASH was a project funded in part by the CANARIE Learning Program [12]. SPLASH was originally built on Sun Microsystems JXTA platform but latter replaced JXTA with its own peer-to-peer system that is more efficient and capable of communicating across firewalls. SPLASH was the exploratory research of ECL. Experiences gained from the development of SPLASH, notably in the metadata and LOM query mapping, and had transcended onto the development of the ECL infrastructure.

8. Conclusion

This paper is based on the premise that the LOR network is a collaborative and evolving environment, and the reusability of LOs is the fundamental driving idea behind the LOR network. Therefore, the need for a flexible, yet stable communication infrastructure is imminent to ensure the ongoing development of e-learning technologies. Middleware technology gives ECL a clear technological advantage regarding the integration of the ECL infrastructure. However, to be successful, ECL must continue working development of the ECL and continue provide support to developers. Finally, if developed according to the interoperability framework specifications, SQI will

need to use similar technologies as ECL to achieve the same advantages. However, having in mind the backing that SQI has, SQI could have a bright future.

9. References

- [1] Y. Atif, R. Benlamri and J. Berri, "Learning Objects Based Framework for Self-Adaptive Learning," *Education and Information Technologies*, vol. 8, pp. 345-368, 2003.
- [2] M. J. Bass and M. Branschovsky, "DSpace at MIT: Meeting the challenges," in *Proceedings of the 1st ACM/IEEE-CS Joint Conf. on Dig. Lib.* 2001, pp. 468.
- [3] M. Hatala, G. Richards, T. Eap and J. Willms, "The eduSource communication language: Implementing open network for learning repositories and services," in *SAC '04: Proceedings of the 2004 ACM Symposium on Applied Computing*, 2004, pp. 957-962.
- [4] S. Encheva and S. Tumin, "Cooperative Shared Learning Objects in an Intelligent Web-Based Tutoring System Environment," *Lecture Notes in Computer Science*, vol. 3675, pp. 227-234, 2005.
- [5] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," in *IFIP International Conference on Network and Parallel Computing*, 2005, pp 2-13.
- [6] R. Kazman, L. Bass, M. Klein, T. Lattanze and L. Northrop, "A Basis for Analyzing Software Architecture Analysis Methods," *Software Quality Control*, vol. 13, pp. 329-355, 2005.
- [7] J. Kontio, G. Caldiera and V. R. Basili, "Defining factors, goals and criteria for reusable component evaluation," in *CASCON '96: Proceedings of the 1996 Conference of the Centre for Advanced Studies on Collaborative Research*, 1996, pp. 21.
- [8] W. Kraan, "Issues in Federating Repositories: A Report on the First International CORDRA™ Workshop," *D-Lib Magazine*, vol. 11 No. 3, March 2005.
- [9] M. Lindvall, R. T. Tvedt and P. Costa, "An Empirically-Based Process for Software Architecture Evaluation," *Empirical Softw. Engg.*, vol. 8, pp. 83-108, 2003.
- [10] V. Reklaitis, K. Baniulis and T. Okamoto, "Shaping e-Learning Applications for a Service Oriented Grid," In *Proceedings of 2nd International LEGE-WG Workshop on e-Learning and Grid Technologies*, 2003.
- [11] V. Reklaitis, K. Baniulis and A. Masevicius, "Towards e-learning application architecture based on GLOBUS framework," in the *Proceedings of Euroweb 2002 Conference*, 2002.
- [12] G. Richards, M. Hatala and R. McGreal, "POOL, POND and SPLASH: Portals for Online Objects for Learning," In McGreal, R.(ed): *Online Education Using Learning Objects*, RoutledgeFalmer, London, 2004, pp. 236-243.
- [13] B. Simon, D. Massart, F. V. Assche, S. Ternier and E. Duval, "A simple query interface for interoperable learning repositories," 2005, Available at: <http://nm.wu-wien.ac.at/e-learning/interoperability/www2005-workshop-sqi-2005-04-14.pdf>.