# The Interoperability of Learning Object Repositories and Services: Standards, Implementations and Lessons Learned

Marek Hatala, Griff Richards[1], Timmy Eap and Jordan Willms
School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC, Canada, V3T 2W1
+1-604-268-7431
{mhatala, griff, teap, jpwillms}@sfu.ca

## ABSTRACT
Interoperability is one of the main issues in creating a networked system of repositories. The eduSource project in its holistic approach to building a network of learning object repositories in Canada is implementing an open network for learning services. Its openness is supported by a communication protocol called the eduSource Communications Layer (ECL) which closely implements the IMS Digital Repository Interoperability (DRI) specification and architecture. The ECL in conjunction with connection middleware enables any service providers to join the network. EduSource is open to external initiatives as it explicitly supports an extensible bridging mechanism between eduSource and other major initiatives. This paper discusses interoperability in general and then focuses on the design of ECL as an implementation of IMS DRI with supporting infrastructure and middleware. The eduSource implementation is in the mature state of its development as being deployed in different settings with different partners. Two applications used in evaluating our approach are described: a gateway for connecting between eduSource and the NSDL initiative, and a federated search connecting eduSource, EdNA and SMETE.

## Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**] Systems and Software – *information networks* H.3.7 [**Information Storage and Retrieval**] Digital Libraries – *standards, systems issues, dissemination*. D.2.12 [**Software engineering**] Interoperability

## General Terms
Design, Standardization

## Keywords
Learning object repositories, Interoperability

## 1. INTRODUCTION
In the last few years we have seen a significant progress in the area of crucial technologies and standards for the semantic web: XML and RDF have gained wide acceptance in the industry, and the Semantic Web group at W3C is finalizing the recommendation for next essential semantic web component the Ontology Web Language. Metadata are in use across all vertical layers of the systems and several large-scale initiatives are trying to build usable networked systems for object and knowledge sharing and to further our understanding of the related issues. All these activities promise to have systems that can discover and share information with other systems in place in the near future.

One of the leading areas where integration and sharing is in high demand is education, particularly in e-learning. The wholesale adoption of Internet technology as a channel for education and training has resulted in an abundance of learning resources in web-ready digital format. Typically, these digital learning objects (LO) [15] may be lesson content stored as text, audio-visual or interactive media files, or simply learning activity templates expressed in a learning design format [7]. Despite their apparent ubiquity, the locating and re-use of LOs is hampered by a lack of coordinated effort in addressing issues related to their storage, cataloguing and rights management. Strident efforts have been made to create portal repositories by communities such as Merlot, SMETE and, in Canada, by TeleCampus and CAREO. Not surprisingly, each entity produces a rather individual reflection of its own perceived organizational needs, and the concept of making all these repositories work together while laudable, has received less attention.

The e-learning community has seen fruitful initiatives in the standardization of learning object metadata by IEEE and the emergence of specifications towards the standardization of other aspects of learning objects and learning processes by organizations such as IMS and ADL. More recently, the e-learning community has been focusing on the ability to connect and use resources located in distributed and heterogeneous repositories. This process closely resembles the initiatives in the domain of digital libraries, to the extent that there are initiatives such as the recent Alt-i Lab meeting at MIT to bring these two communities together. In the next section we examine how the interoperability is handled in four major projects: the National Science Digital Library project, the IMS Digital Repository


*WWW 2004*, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

[1] BCIT Technology Center. c/o School of Interactive Arts and Technology, SFU.

Interoperability group, our recent POOL project, and our current approach to interoperability in the eduSourceCanada project [11]. In Section 3 we describe the main drivers of the eduSource project as an infrastructure for connecting different types of networks and people. This provides us with main issues when creating large and open networks. In Section 4 we discuss the eduSource architecture, eduSource Communication Layer and enabling middleware for easy connection to the network and between eduSource and other networks. In Section 5 we give a current status of the implementation and describe two uses cases providing validation of our approach.

## 2. MAJOR INTEROPERABILITY EFFORTS IN E-LEARNING

***OAI.*** Although not specifically oriented to education, the Open Archive Initiative [13] develops and promotes interoperability standards for *content dissemination.* The Protocol for Metadata Harvesting (PMH) developed by OAI provides an application-independent interoperability framework for metadata harvesting. The protocol enables repositories (called harvesters) to selectively harvest metadata from other sources (providers) and create cumulative and/or specialized collections of metadata. In addition to the protocol, OAI provides guidelines and community support. The protocol is used widely used by other initiatives to support harvesting functionality.

***NSDL.*** The National Science Digital Library project (www.nsdl.org) is a major project funded by the National Science Foundation with the goal of building a digital library for education in science, mathematics, engineering and technology. The potential collections for inclusion in NSDL have a wide variety of data types, metadata standards, protocols, authentication schemes, and business models [1]. The aim of the NSDL interoperability is to build coherent services for users from technically different components. NSDL aims to support three levels of interoperability:

1. ***federation*** implements the strong standards approach with libraries agreeing to use specific standards.
2. ***harvesting*** allows higher autonomy. The only requirement is to enable a limited set of services via a simple exchange mechanism. NSDL is using Protocol for Metadata Harvesting (PMH) developed by the Open Archive Initiative. Harvesting is supported on the repository side by implementing a relatively simple wrapper communicated via PMH and providing metadata based on Dublin Core.
3. ***gathering*** uses the web crawler technique to collect information from the organizations that do not formally participate in the NSDL program.

NSDL has selected eight preferred metadata element sets for metadata storage. While member libraries can store the metadata in their original local format, they have to be able to serve the metadata in Dublin Core (www.dublincore.org) format. Effectively this solution establishes Dublin Core as the lowest common denominator for the NSDL.

***IMS DRI.*** The IMS Digital Repository Interoperability Group, in its specifications for the digital repository interoperability [6], provided a functional architecture and reference model for repository interoperability. Aiming at very broad application of the specification the DRI document makes recommendations only to a certain level and leaves the resolution of more operational issues to the system implementers. Five basic functions defined by IMS DRI are: search/expose, gather/expose, submit/store, request/deliver, and alert/expose. For the *search* function, the specification recommends using either XQuery (www.w3c.org/XML/Query) with SOAP protocol or Z39.50. For the *gather* function, the OAI's harvesting protocol is recommended. No recommendation is made for the other three functions in the current version of the specification. The current version of IMS DRI envisions but does not explicitly deal with heterogeneity of the repositories and it is up to the implementers to ensure format compatibility. The DRI Group recommends development of "search intermediaries" that will deal with multiple formats.

***POOL.*** The POOL project ran from 1999 to 2002. One of its major goals was to build an infrastructure for connecting heterogeneous repositories into one network [5]. The infrastructure used a peer-to-peer model in which nodes could be individual repositories (called SPLASH) or community or enterprise repositories (PONDs). PONDs were connected to the POOL network using a specialized peer performing the functions of both a gateway and wrapper. The POOL network used the JXTA peering protocol (www.jxta.org) and followed the CanCore/IMS metadata profile/specification (www.cancore.org) to exchange metadata. Connected PONDs communicated using wrappers either via HTTP and CGI or XML-RPC protocol. The wrapper also performs the metadata schema translation functions that are needed. The network supported high autonomy for the repositories, but this required creating a specialized wrapper to translate between the metadata schemas and communication protocols.

***ELENA/Edutella.*** This collaborative European project is creating Smart Spaces for Learning [12]. Smart learning spaces are defined as educational service mediators, which allow the consumption of heterogeneous learning services via assessment tools, learning management systems, educational (meta) repositories and live delivery systems such as video conferencing systems. ELENA builds a dynamic learner profile which is used as a basis for offering the learner with the choice of a variety of knowledge sources. ELENA forms a layer on top of a learning management network built on Edutella [14]. Edutella is an RDF based peer-to-peer (P2P) infrastructure that aims to connect highly heterogeneous educational peers with different types of repositories, query languages and different kinds of metadata schemata.

***eduSourceCanada.*** The eduSource project (www.edusource.ca) brings together major Canadian LOR players to create an open infrastructure for linking interoperable LORs. The infrastructure will support a wide range of services and promises both ease of connecting and ease of using new and existing systems. For example, a repository using PMH protocol and Dublin Core metadata can either communicate with the eduSource network as a whole via the gateway mechanism or it can become a participant with access to wider range of services via the ECL interoperability connector.

***OKI.*** The Open Knowledge Initiative [9] builds an open and extensible architecture that specifies how the components of an educational software environment communicate with each other and with other enterprise systems. The OKI provides a service specific API called Open Service Interface Definition (OSID) that fosters an effective *application development* for higher education by providing definitions for data and common services. OSID covers a wide range of learning services from generic ones such as Authentication and Digital Repository to services specific to education such as Course Management and Grading. Currently OSID has few test implementations but has promising support both from the academic and industrial community.

## 3. IMS DIGITAL REPOSITORY INTEROPERABILITY

IMS is a global learning consortium developing specifications for a wide range of learning contexts. The specifications range from individual learning resource metadata specification, through specific learning specifications such as a question and test interoperability to more generic digital repository interoperability specification (IMS DRI). IMS specifications are defined by three documents: the information model, the XML binding and the best practices implementation guide.

The first version IMS DRI specification (released in January 2003) provides recommendations for the interoperation of the most common repository functions. The specification does not make any assumptions about the repositories and treats them as a collection of resources. The IMS DRI information model defines 8 core functions (see Table 1) where 3 are defined at the repository level (store, expose, deliver) and 5 are defined at the 'resource utilizer' level (search, gather, submit, alert, request).

The reference model in the specification provides recommendations on functional architecture and for specific technology. However, the level of recommendations is very high leaving many specific details unanswered. The following five specifications are used for the combination of core functions:

1. Search/Expose. Two query languages are recommended: XQuery for XML format and Z39.50 for searching library information.
2. Gather/Expose. No specific recommendation is made; the IMS DRI suggests that the OAI model will provide a sufficient functionality.
3. Alert/Expose. No specific recommendation is made as Alert is regarded out of scope of first version of the recommendation.
4. Submit/Store. The specification recommends using the IMS package as a SOAP attachment.
5. Request/Deliver. The specification excludes several related issues from its scope leaving implementers with the general guidance of using http and ftp for different types of resources.

**Table 1 eduSource services**

| ECL Service | Description |
|---|---|
| *Expose* | When asynchronous messaging is required, this service will be called by service providers to return the responses for search, gather, and alert. |
| *Gather* | Repositories wanting to provide gather service must implement gather service handler. |
| *Search* | As recommended by IMS DRI, ECL protocol uses XQuery. To enable connection of the repositories that do not support (full) XQuery a set of XQuery templates is used. The repositories register their Search Service with an indication of supported templates or full XQuery search capability. |
| *Alert* | IMS DRI recommends Alert for push gather. Whenever repository has new metadata matching subscribe parameters, it sends an alert message to the subscribers. |
| *Submit* | It is a function for moving an object (metadata and learning object) to a repository. |
| *Store* | When asynchronous messaging is required, this service is called by service providers to return the results of submit function. |
| *Request* | It provides a function to ask to deliver objects to a client. The transfer protocol could be a successive SOAP request to download the object or FTP transfer protocol |
| *Deliver* | When asynchronous messaging is required, this service will be called by service providers to return the chunked of results. |

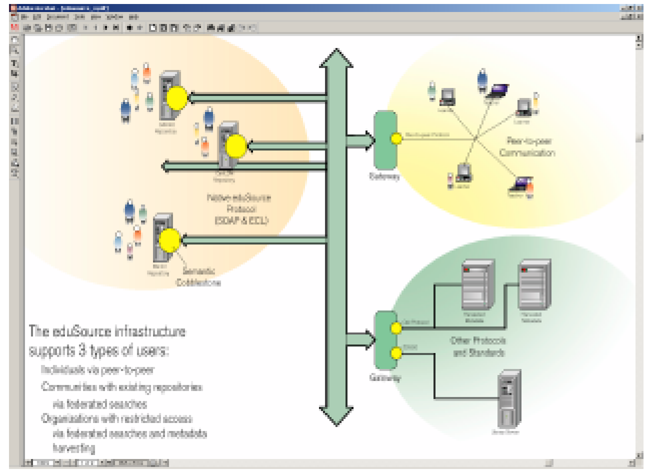


**Figure 1. EduSource infrastructure supports three types of communities**

The IMS DRI Core Functions XML binding document specifies a SOAP messages over HTTP protocol as an initial message binding and defines the general message structure. Once again, the specification is not very 'specific' and leaves many detailed questions open.

## 4. EDUSOURCE: AN OPEN NETWORK FOR CONNECTING COMMUNITIES

To achieve its goals the eduSource project is implementing the IMS DRI specification as closely as possible. To understand eduSource's strong requirements for interoperability we need to analyze the reality of the education space and the variety of communities that eduSource will serve.

***Server-type repositories.*** Figure 1 shows a schematic infrastructure of eduSource network. The top left quadrant represents *server-type repositories*. The communities served by these repositories vary and can include governmental, academic, business or special interest groups. Some of these repositories were created and are managed by an organization expressly to serve their communities. For example, a university repository primarily serves its community of university students and professors; similarly a provincial Ministry of Education might operate a repository of learning resources for K-12 schools within its jurisdiction. Another type might be a commercial repository that licenses their content or charge fees per use. Another common type is informal repositories that are not tied to any formal organization but were simply set up by the community members themselves and are managed to further the community goals. In all these cases the repositories can be either public, or restricted to serve only the community, or can provide mixed access with a blend of privileges depending upon user identity and role.

The server-type repositories generally provide access to their functionality through a web portal. This includes search and create functionality for metadata and view functionality for the resources. The metadata schema is determined by the repository developer and cannot be easily changed. One example of an interesting repository is CAREO which in addition to web forms for metadata creation also provides a specialized application, ALOHA, for metadata creation and uploading of metadata and learning objects to the appropriate repository.

***Peer-to-peer Repositories.*** The top right part of the Figure 1 represents a network of individual repositories which communicate on a peer-to-peer (P2P) basis. SPLASH [5] is an example of a P2P repository that was developed in POOL. Individual SPLASH repositories provide the storage and management functions for the learning objects used or collected by an individual user. SPLASH also enables its users to create metadata for the learning objects residing either on the individual's file system or on the web. SPLASH uses P2P protocol to search for learning objects on other peers[2] and provides file swapping functionality to transfer learning objects between peers.

Peer-to-peer repositories serve the needs of the individual instructors and learners who may not have centralized repository support from their organizations. They are also of preference for those who object to the loss of control over their resources and imposed limitations when using centralized repositories, and they can serve as test sites for objects under construction either by content authors, or as products of constructive learning activities. P2P repositories enable each individual to be included and contribute towards the community resources with minimal technical requirements. P2P repositories may lack the system support of the server-type-repositories but they often provide their users with additional object management functions and facilitate cross-repository searches. A side benefit typical of P2P systems is their potential scalability when high demand for a particular type of object occurs.

***Repositories of Harvested Metadata.*** Metadata harvesting is an alternative to federated searches - instead of constantly sending search requests out to all the primary repositories, harvesters collect metadata into a centralized location and searches scan the centralized collection. To be efficient, a search engine might harvest metadata from previous searches, and only conduct new searches when necessary. In another scenario, a harvester might continually poll repositories for new metadata records. Harvesting works well for repositories that use the same or easily-mapped metadata schemas as the queries are typically specified in one schema only.

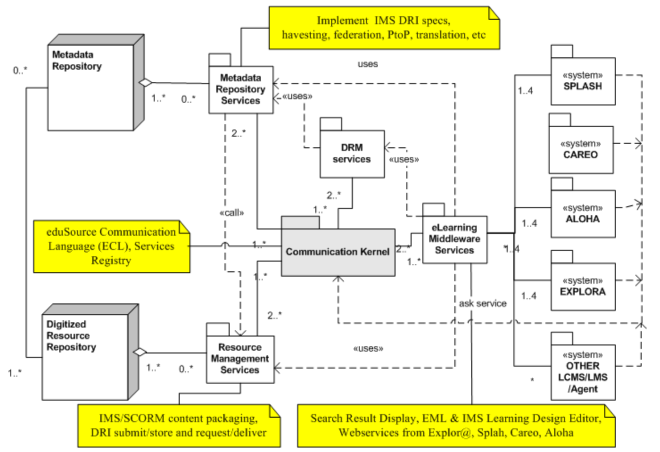


**Figure 2 eduSource functional architecture**

It is important to note that not all primary sources (repositories) allow harvesting of their metadata. This is especially true of commercial repositories where their business model depends on the users visiting their repository directly. Some repositories only allow harvesting of certain metadata fields. In general, proprietary repositories prefer federated searches which generate results that direct potential users to the company's own website.

***External Repositories and Networks.*** eduSource places an emphasis on connecting to other significant initiatives and networks. These connections can be bi-directional, enabling both eduSource users to search beyond the eduSource network and external users to find resources inside eduSource. Alternatively an external repository can use our preconfigured middleware to connect to the eduSource network.

# 5. ECL: EDUSOURCE COMMUNICATION LAYER

A communication protocol plays an important role in each of the major initiatives listed in the previous section. It allows members of the initiative to achieve its goals by allowing communication between its members, tools, and services. EduSource is a broad network as it aims at the wide spectrum of services it wants to support. On the other side, for eduSource to become an open network it has to build its protocol on existing standards and recommendations. EduSource defines its *eduSource Communication Layer* (ECL) as an implementation of the IMS DRI specification. However, as noted in Section 3, the IMS DRI recommendation is not specific enough for direct implementation and the current penetration of recommended technologies is not as widespread as assumed in the specification. This makes the ECL implementation a challenging and exciting endeavor.

## 5.1 General approach

The eduSource architecture uses the web services approach in which services communicate using the ECL protocol [10] (Figure 2). Although choosing the web services approach was a straightforward decision, selecting associated technologies needed

[2] In the POOL network SPLASH also searches server types repositories which were connected to the POOL network. In the eduSource network this functionality is being replaced by the more generic ECL approach.

more consideration. The criteria for the protocol and its development process that affected our approach included:

- eduSource is a heterogeneous network consisting of existing and future institution repositories, peer-to-peer network, individual small repositories, and application interfaces.
- ECL will be evolving over the time of the project which makes all the parallel activities vulnerable to changes in the protocol.
- ECL supports many new services non-existing in the current systems. Some of these services require asynchronous communication, such as search through peer-to-peer network or alert.
- ECL is a complex protocol. To achieve significant adoption, it has to be fast and easy to use and be supported with pre-configured middleware.
- A solution for connection between eduSource and other initiatives has to be easy to maintain and easy update if there is a change in the protocol used by the other initiative.

After thorough development team discussions we opted for the document style web services [8][4] over SOAP (see Section 7 for detail discussion on the issue). ECL closely follows the IMS DRI specification and uses SOAP as a communication layer. IMS DRI core functions (Table 1) are defined and implemented as eduSource services. Repositories or tools connected to the eduSource network can implement some of these services and register them in an eduSource maintained registry (such as UDDI). Registration is a preferred way for discoverability of permanent services. However, in many cases user tools connected to the network will not register any service[3]. This was made possible by declining RPC-style web services approach as the only way of implementing services in eduSource.

ECL is a complex protocol with communication patterns that may be challenging to implement (such as asynchronous communication). To lower the technical barriers for service providers to join the network it was essential to have a solution that made the ECL easy to implement. Thus, as we develop the ECL we are building the eduSource connector - a middleware that exposes eduSource services in the form of handlers and hides all the complexity of properly encoding XML messages and communicating with other eduSource services.

## 5.2 ECL Connector

Since the complexity of the ECL protocol might be detrimental to its adoption, we are providing an "off-the-shelf" eduSource connector which implements the ECL protocol. The connector provides a standard API to connect an existing repository to the eduSource network. The ECL protocol only requires institutional repositories or repository tools to implement the connector handlers for those specific services they want to expose to others (Figure 3). This is simpler than implementing and deploying every service in each institution. The connector also facilitates version synchronization during the protocol evolution. Changes in the protocol itself rarely propagate to the API level. In most cases, repositories do not have to worry about the change in the

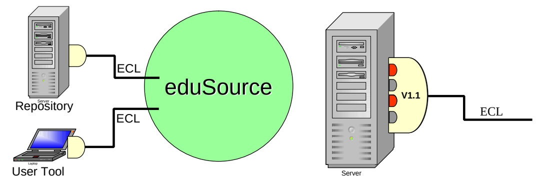


**Figure 3. EduSource Connector**

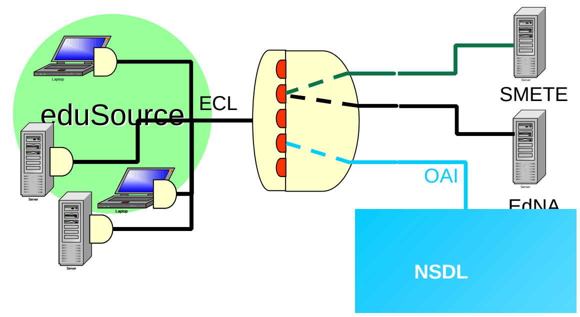


**Figure 4. EduSource Gateway**

protocol; they only need to update the connector newer versions. Changes in the ECL protocol can be detected by the newer version of the connector and are dealt with automatically. This feature makes the implementation of the ECL protocol very attractive, especially in this early development stage where the implementation is still evolving.

The connector hides the complexity of the communication between two eduSource nodes. This is especially advantageous in the case of the asynchronous messages that are difficult to implement.

It is important to note that an API is language specific while a protocol is language agnostic. While the ECL connector with its API simplifies the connection process for those working in the same language, e.g. Java, describing the protocol provides an opportunity for different programming language communities to implement and share ECL connectors in their preferred language[4].

## 5.3 ECL Gateway

Although the eduSource internal protocol provides a flexible and efficient solution it is unlikely that well established repositories and initiatives will invest resources and convert their protocols to the ECL. Thus an ability of the eduSource project to connect to other established protocols and major initiatives is of the utmost importance to the project participants. EduSource addresses the problem of outside interoperability by providing a second type of mediator simply called the eduSource Gateway. The eduSource Gateway is modeled after the design pattern of an adapter [3] functioning at the network level. The main function of the gateway is to mediate between ECL and communication protocols used by the outside systems.

Figure 4 shows a schema of the eduSource Gateway. One side of the gateway is formed by the ECL connector. The other side of the gateway provides a framework (Figure 5) defining a chain of

[3] For example, a search application does not provide any services on its own but needs to implement 'deliver' service for asynchronous search results.

[4] At the time of writing a stable version of the Java and partial version of Python connectors are available.

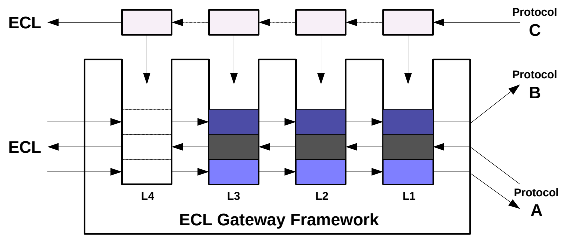


**Figure 5. EduSource Gateway Framework**

handlers that perform a conversion between ECL protocol and the protocol of the external network. The gateway framework enables us to define the mapping between protocols at four levels:

L1. Communication protocol (HTTP, SOAP, XML-RPC, Peer-to-peer, etc.)

L2. Communication language (ECL, OAI, POOL, etc)

L3. Metadata (IMS, CanCore, Dublin core)

L4. Ontologies (vocabularies for metadata)

The eduSource Gateway is typically running on a dedicated computer and provides services for all participants in the eduSource network. The main benefit of placing the mapping functionality for an outside network onto a gateway instead of with each participant is that it can be easily updated if the change in the outside network protocol occurs. In such a case, a chain of mapping handlers is updated at the one place and all eduSource participants can continue to communicate with the gateway using ECL protocol without any change necessary. There can be several gateways for the same outside network if the traffic between the two networks is high. One gateway can provide services for several other networks.

The gateway also functions as a selector of internal eduSource services for the external requests. Currently, any request addressed directly to an eduSource node is forwarded to that node, while a request addressed to eduSource as a whole is distributed to all registered nodes providing the requested service. This is not the best way of distributing the requests and needs to be further addressed.

# 6. IMPLEMENTATION

Designing and implementing the eduSource architecture including ECL is just one of several goals that the eduSource project is attempting to achieve in a short period of 18 month with a $8M budget. This intense project requires simultaneous advancement of several interconnected development activities. We have opted for an iterative development process which saw the most important, most informing and riskiest functionality implemented in the first round. Our aim was to develop the frameworks for the connector and gateway for the most complex services. This enabled us to test the feasibility of our approach while address other issues such as the critical need for content in the eduSource network.

We have opted for the 'gather' function as our first ECL service. As IMS DRI recommends using the OAI protocol we have implemented the OAI protocol commands into ECL communicating via SOAP messages. Gather functionality enabled us to implement first versions of both the connector and gateway. The gateway for mapping between ECL and OAI enabled us to connect to the large NSDL initiative that provided eduSource participants with access to the extensive NSDL resources.

The search functionality was the second ECL service to be implemented. Again, we have implemented search service in the connector and connected existing systems and tools used by eduSource partners. The search service in eduSource is peculiar as we are connecting server type of repositories via federated search and we connect to our SPLASH peer-to-peer system that uses a broadcast-relay search mechanism. To connect to the peer-to-peer system we have implemented another gateway translating between ECL and peer-to-peer protocol. To test the flexibility of our solution we have took a challenge to implement a federated search similar to the one presented by Merlot team at the Merlot 2003 conference [2]. We were able to implement two chains of handlers for EdNA (http://www.edna.edu.au) and SMETE (http://www.smete.org) into an existing ECL-OAI gateway and build a simple web based interface in 2 weeks time with one part-time programmer. We have also implemented a gateway between ECL and SRW (http://www.loc.gov/z3950/agency/zing/srw/) protocol used by digital library systems. The federated search experiment is accessible available from the http://www.edusplash.net website.

The implementation of the 'submit' service required us to expand our SOAP implementation framework with SOAP attachments and the introduction of IMS Packaging for submitting object and metadata to the repository. Another aspect of submit is that many repositories require some level of authorization for the submit operation. ECL does not provide a user registration service; instead the user registers with the specific repository directly. ECL uses a PKI mechanism to secure sensitive login information in the submit messages.

## 6.1 Connector Implementation Example

As mentioned above, in eduSource the ease of connecting an existing system is supported by a middleware component, the ECL connector. The connector provides a developer with a well defined API hiding the complexity of the ECL implementation. Each core function is represented in the API by a handler. Once developers decide what functionality their repository or tool needs

```
public abstract class SubmitServiceHandler
        implements ServiceHandler{

  public SubmitServiceHandler(){}
  public EclMessage digest(EclMessage eclMessage,
DataHandler dh){
    try{
      Submit request =
              new SubmitImpl(eclMessage.getPayload());
      Store response = processSubmit(request, dh);
      eclMessage.setMessageType(EclMessage.STORE);
      eclMessage.setPayload(response.toXmlElement());
    }catch(SubmitException e){
      eclMessage.setMessageType(EclMessage.ERROR);
      eclMessage.setErrorMessage(e.getMessage());
      eclMessage.setPayload(null);
    }
    return eclMessage;
  }
  public abstract Store processSubmit(Submit request,
DataHandler dh) throws SubmitException;
}
```

**Figure 6. SubmitServiceHandler from ECL connector**

```
public class SubmitServiceHandlerImpl extends
SubmitServiceHandler{
  RepositoryBean rb;
  public SubmitServiceHandlerImpl(){
    rb = new RepositoryBean();
  }
  /*  init allows instantiator to initialize this
class with user input parameters */
  public void init(Properties params){
  }
  public Store processSubmit(Submit request,
DataHandler dh) throws SubmitException{
    try{
      String transId = rb.saveImsContent(
        request.getTransactionId(),
        request.getUsername(),
        request.getObjectAccessPermission(),
        request.getImsContentName(),
        dh.getInputStream(),
        request.isUnpack());
      Store store = new StoreImpl();
      store.setTransactionId(transId);
      return store;
    }catch(Exception e){
      e.printStackTrace();
      throw new SubmitException("Unable to process
submit");
}}}
```

**Figure 7. A tool specific Submit implementation**

they implement the appropriate handler that bridges their system with the eduSource world.

Figure 6 illustrates how an ECL handler can be implemented. This example, the Submit handler, was chosen to show the complexity of file attachment.

ECL provides an abstract class (SubmitServiceHandler), which defines the method (`processSubmit`) that needs to be implemented. The digest method processes the ECL message and makes the call to the abstract method processSubmit. The "request" parameter is an instance of Submit class, which contains all the submit parameters. The "dh" parameter is the instance of Java Activation Framework DataHandler class. It holds the data stream of the file attachment. The return value of processSubmit method should be an instance of Store class, a structure provided by ECL for Submit response to inform the caller about the Submit request status.

Figure 7 shows an example of SubmitServiceHandler implementation with the developer supplied code in bold. As defined in ECL protocol, the file attachment is IMS content package in zip format. In this example, the repository requires transaction id, username, object accessing permission type, the IMS content package name, the input stream of the zip file, and the deployment instruction as parameters. The transaction id is a ticket that allows client to update and delete the submitted content.

## 7. DISCUSSION

In this section we discuss several challenges that we faced in the design and implementation of the interoperability mechanism. This is especially true considering it is one of the first implementations of a specification that is not well articulated and recommends technology (i.e. XQuery) that is not widely used in the real world applications. On the other side, early implementations allow us to further inform the specification process and provide best practice recommendations. Implementation and deployment also provide an opportunity to discover the synergies with other approaches and to define next set of questions that needs to be addressed through the specification process to achieve a higher level of interoperability.

### 7.1 Pragmatics of Following the IMS DRI

The following of recommendations from IMS DRI required making several pragmatic decisions. One major obstacle we faced was following the recommendation for using XQuery as a query language for search functionality. The reality is that there are very few products that currently support XQuery, indeed, many of the existing repositories of eduSource stakeholders do not support XQuery. Two possible solutions to address this problem were:

1. degrade query language to a less powerful but commonly supported language, such as XPath; or
2. use XQuery but provide a solution that will enable all repositories to participate in eduSource.

Although the first option looked expedient, we opted for the second option mainly because of the potential long-term benefits of having a solution following the recommendation from IMS. We have implemented several template XQueries to satisfy the requirements of the major stakeholders[5]. Participants without XQuery support may implement as many templates as they want to support and register these services with their explicitly specified supporting format. EduSource participants who support the full XQuery will support all defined templates through their XQuery engine.

### 7.2 Document-style Web Services

As mentioned in Section 5.1 above we he have chosen a document-style web services approach over the more commonly used remote procedure call (RPC) style. RPC style is more common as people started to implement web services using the familiar paradigm from RMI and CORBA for exposing server-side data and functions. The RPC method can also be easily supported by frameworks and tools. On the other side, document-style web services offer a satisfying mix of well-defined structures and interoperability [4]. The benefits of document style for the development of complex interoperability protocols, such as ECL include full use of XML, ability to validate request and objects using XML-schemas, and making object exchange more flexible.

In case of ECL, the ECL protocol is much richer than the framework expressible by RPC calls that require a rigid contract. RPC calls do not provide enough coverage for eduSource as a heterogeneous network. Specifically, ECL implements a whole set of new services that are now possible, such as 'push gather', 'subscribe', 'alert', while RPC-style makes the support for the asynchronous messaging difficult to achieve. Connecting peer-to-peer networks into eduSource also needs asynchronous messaging as the search results from the broadcasted search will become available in batches. This is also true when processing large amounts of data where it is more manageable to have asynchronous messaging deliver results in batches. Document-style web services also make object exchange easier and allow making full use of XML. This is essential as we have to deal with

[5] Templates differ by their query capabilities and ways how they format their results. For example, one template specifies keyword based search and formulates results in brief format. Another template specifies keyword search and returns full IEEE LOM records.

the reality that, at best, different repositories will use different variants of the metadata standards or, at worst, completely specific metadata that needs to be mapped to the standards or transferred unchanged.

### 7.3 Comparison with Other Approaches

Section 2 listed relevant projects and approaches to interoperability in the current learning objects arena. They can be compared using different criteria.

- From the perspective of scope of functionality the POOL project, Edutella/Elena, eduSource and OKI project all aim at supporting a wide range of interoperability functions.
- The NSDL meta-repository initiative is too broad to be easily categorized but does have several goals overlapping with our project.
- The OAI project focuses on dissemination - clearly addressing one part of eduSource goals. The approaches differ at the level they address the problem of interoperability.
- POOL and Edutella projects used peer-to-peer idea to connect repositories and tools with different capabilities.
- Elena project development is at an early stage with early drafts suggesting that the project will not follow the IMS DRI specification as closely as eduSource has. Another difference between two projects is that Elena is using web services approach with remote method invocation approach (see above) with all the consequences of tightly coupled system.

The comparison with OKI OSID is interesting as OSID aims mainly at the interoperability between components within a learning systems during its development while ECL addresses the problem of interoperability between standalone learning systems and services. This makes the two approaches complementary, and we are exploring potential linkages[6].

### 7.4 Future Development

The ECL in its current stage of development and implementation is a stable interoperability mechanism that is currently being deployed within Canada and internationally. In Canada, the repositories and tools are being connected into the eduSource network expanding a suite of eduSource enabled tools available for the download and wide adoption by the learning community. The interoperability mechanism has gained an interest from the industrial community as well with more than a dozen industrial partners currently working with eduSource on adoption of ECL interoperability for their products.

We are discussing a possibility to link with other international initiatives in the area of learning objects and digital libraries. In Europe, the ECL is being considered for adoption by the UK CETIS group and there are discussions to integrate with the European SchoolNet through the Celebrate project.

On the research side, our group is a partner in the LionShare project funded by the American A.W.Mellon Foundation. LionShare started in the fall of 2003 and its goals include integration of ECL with OKI OSID, embedding Shibboleth (the Internet 2 distributed trust mechanism) into the ECL connecting middleware and investigating bridging between heterogeneous peer-to-peer networks. Other projects funded through the Canadian LORNet Research Network look at extending the ECL to the semantic web, and extending the functionality of the SPLASH Network to include other types of learning objects such as Learning Designs, and e-portfolio artifacts.

## 8. CONCLUSIONS

In a perfect world there would be only one metadata protocol and we would need only one repository and one search mechanism. However, this would be a rather bland world. The reality of e-learning is a hodge-podge of legacy repositories, protocols, special interest groups and self-serving communities. Rather than preach conformance, the eduSourceCanada project focused on the common functions desired by the owners and user of learning object repositories and strived to intermediate between the technologies involved. Our previous experience with POOL, POND and SPLASH proved that heterogeneous repository types could and should co-exist and serve a global interest in the re-use of learning objects. Although the protocols described in this paper are but baby steps in that direction, they demonstrate that the technical barriers can be overcome and that robust solutions to interoperability are not far away. However, the ultimate challenges to interoperability remain political – we can only interoperate with those repositories that wish to do so.

## 9. ACKNOWLEDGEMENTS

The authors acknowledge their valuable discussions with Jean-Francois Arcand in conceptualizing the ECL. The eduSource work at SFU Surrey is partly funded through the Canarie Inc. eLearning Program, and the NSERC LORNet Research Network.

## 10. REFERENCES

[1] Arms, W. Y., Hillmann, D., Lagoze, C., Krafft, D. , Marisa, R., Saylor, J. , Terrizzi, C. and Van de Sompel, H. (2002) A Spectrum of Interoperability: The Site for Science Prototype for the NSDL, *D-Lib Magazine*, 8 (1), January. http://www.dlib.org/dlib/january02/arms/01arms.html

[2] Bastiaan, M.K., Lalanne, A., Shamseldin, S. MERLOT Federated Search Technologies. Presentation at the Merlot International Conference, Vancouver, Canada, August 5-8, 2003. (http://fedsearch.merlot.org)

[3] Gamma, E., Helm, R., Johnson, R.and Vlissides, J. (1995) Design Patterns, Reading, MA: Addison-Wesley

[4] Gibbs, K., Goodman, B.D. and Torres, E. Create Web services using Apache Axis and Castor, IBM developerWorks, http://www-106.ibm.com/developerworks/webservices/library/ws-castor/ (accessed Nov 14, 2003)

[5] Hatala, M. and Richards, G. Global vs. Community Metadata Standards: Empowering Users for Knowledge Exchange, In: I. Horrocks and J. Hendler (Eds.): The Semantic Web – ISWC 2002, Springer, LNCS 2342, pp. 292-306, 2002.

[6] By simple analogy, the OSID might be thought of as specifications for plumbing fittings inside a house, while the ECL provides methods for connecting the house to the water distribution system.

[6] IMS DRI (2003) IMS Digital Repositories Interoperability - Core Functions Information Model, Version 1, Available: http://www.imsglobal.org/digitalrepositories/index.cfm

[7] IMS Learning Design Specification, http://www.imsglobal.org/learningdesign/, February 2003

[8] McCarthy, J. Reap Benefits of Document Style Web Services, IBM developerWorks, http://www-106.ibm.com/developerworks/webservices/library/ws-docstyle.html (accessed Nov 14, 2003)

[9] Open Knowledge Initiative, http://web.mit.edu/oki/

[10] Paquette, G. *et all*. The eduSource Suit of Tools: Use Cases Specification (v.06), June 7, 2003, Available: http://www.edusource.ca

[11] Richards, G. and Hatala, M.: Semantic Cobblestones: An Interoperability Mechanism for Learning Object Repositories. In McGreal, R.(ed): Online Education Using learning Objects, Kogan Page, London, 2003 (in press)

[12] Simon, B., Miklos, Z., Nejdl, W., Sintek, M., Salvachua, J. Smart Space for Learning: A Mediation Infrastructure for Learning Services. In Proceedings of the Twelfth International Conference on World Wide Web, Budapest, Hungary, 20-24 May 2003.

[13] Van de Sompel, and H. Lagoze, C. Notes from the Interoperability Front: A Progress Report on the Open Archives Initiative. ECDL 2002, Rome, Italy, Springer, LNCS 2458, , pp.144-157

[14] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmr, and T. Risch. Edutella: A p2p networking infrastructure based on RDF. In *Proceedings of the Eleventh International World Wide Web Conferenc*e, Honolulu, Hawaii, May 2002.

[15] Wiley, D. (ed.) The Instructional Use of Learning Objects, Association for Educational Communications and Technology, 2001