

INTEROPERABILITY FRAMEWORKS FOR LEARNING OBJECT REPOSITORIES

GRIFF RICHARDS

*BCIT Technology Centre.
E-mail: griff@sfu.ca*

MAREK HATALA

*Simon Fraser University, 10153 King George Highway, Surrey, BC CANADA V3T 2W1.
E-mail: mhatala@sfu.ca*

Learning object repositories hold the digital resources that make on-line instruction possible. Whether held by individuals, learning communities, or purveyors of knowledge artifacts, the reusability and hence the potential market for e-learning objects depends on the extent to which the objects can be found, selected for appropriateness, and retrieved for use in a new instructional context. This presentation outlines the current efforts of eduSource Canada, a Canadian consortium building a national interoperability framework for both academic and industrial contexts. As eduSource strives to unite both peer-to-peer and web services models, the mechanisms for interoperability at the transactional and semantic levels are described in some detail. Key to the solutions proposed are the ECL or eduSource Communications Layer, an open protocol to enable search, gather and retrieval within the eduSource community and gateways which extend this functionality to other learning object repository networks.

1 Introduction

As learning objects become more common in the development and deployment of e-learning curricula, there will be increasing impetus to ensure that the learning objects developed for one instructional context can actually be re-used in another. Similarly, there will be increased desire for learning object repositories (LOR), the storehouses of learning objects, to interoperate so that objects stored in one LOR can be discovered, previewed, and transferred to another LOR. Whether this exchange is free or commercial those wishing to exchange objects need to have a clear method of making their collections known and accessible while safeguarding the usage rights and integrity of the objects themselves. This paper discusses the evolution of our LOR work culminating in the outcomes of the eduSource Canada project in implementing systems for the standards-based storage and exchange of learning objects. After providing an overview of interoperability issues and mechanisms, this paper outlines the approach we took to interoperability in POOL, and in the eduSource Communications Layer (ECL). Three possible approaches by which learning object repositories connect to a global network for exchange are explained. The paper concludes with a brief discussion of current work to integrate this framework with the MIT Open Knowledge Initiative and PennState's LionShare. References are provided to more technical discussions and documentation.

2 Background to Learning Object Repositories

Learning objects may be generally regarded to be the digital files that store the content or prescribe the activities for e-learning [5]. There are many variants of the definition [9,14], depending on the size (or granularity) of the objects which may vary from a single image or text block to a complete course, the degree to which the objects fit intrinsically into a reusability scheme, and the intended purpose for the object (either instructional by nature, or re-purposed for instructional ends.) While in conventional computer-assisted instruction, the content and activities might be inseparably hard-coded, in a web-based learning objects environment these elements are generally kept as discrete files that can be inserted or transmitted as required by an instructional plan.

Learning object repositories (LORs) are essentially storage and retrieval systems for learning objects. Given the difficulty of searching non-textual images or applets, learning objects are often catalogued or "meta-tagged" with information about their content, audience and format. Those searching for an appropriate learning object can therefore search a repository's metadata to see if a suitable object is present.

In this sense a learning object repository generally consists of four components: an object store, a metadata creation and maintenance database, a suitable user interfaces and a connection to the Internet.

The problem of linking learning object repositories is not unique to eduSource and many of the fundamental methods integrate specifications from the IMS-Global (www.imsglobal.org)[4]. Acknowledging, subtle variations and combinations, there are four principal architectures for learning object implementations:

1. they can be isolated applications installed on a single computer,
2. they can be configured as a web portal serving a community of learners or of practice
3. they can be set up to “harvest” metadata from several sites onto a single portal as a metadata repository, or
4. they can be a “federation” of linked repositories with each repository storing a subset of the entire collection.

The web portal approach has been favoured by academic portals such as MERLOT (www.merlot.org) [2] and CLOE (www.cloe.ca). Harvesting has been used by third party agencies such as NSDL (National Sciences Digital Library)[1] to organize access to content held by others. The federated approach as used in POOL (www.edusplash.net) [3.11] allows each repository to meet the instructional interests of its owner while enabling the search, discovery and the retrieval of a large variety of objects (to the extent that permissions are granted). A similar peer-to-peer approach has been used in the European Edutella project [12,13] to link community repositories.

Each of these approaches accommodates needs for security and control within the context of community and sharing, and selections are to some degree constrained by the complexity of the configuration and the need for technical know-how to create or configure the repository. Extended communities are more likely to go for a centralized web-portal approach, large organizations for a web services approach and individuals are likely to move towards a federated approach using Napsterish peer-to-peer applications. The ultimate goal, of course, would be to find methods to link all repositories together to make the largest array of objects available to the largest audience of users.

Webster’s Dictionary defines interoperability as the ability of a system (as a weapons system) to use the parts or equipment of another system. The rapid incremental development of technology is possible because developers can build on top of a variety of existing code libraries and standards. Standards play an important role in the development of web technology – Unicode, HTML, and TCP/IP provide important base technologies for the representation of characters, formatting and exchange. The MIT OKI group [7] provides an interesting pyramid analogy for standards – the more base technologies that can be incorporated, the less is left for the individual application developer to invent, and the more effort can be placed on developing the specific application at the tip of the pyramid. With so many e-learning organizations engaging in the learning objects movement, it makes quite good sense to collaborate in the development of standards to improve the interoperability of the emerging e-learning software. This is particularly important for the increasing amount of open source software that needs to engender a sufficient user community to ensure long-term support and evolution.

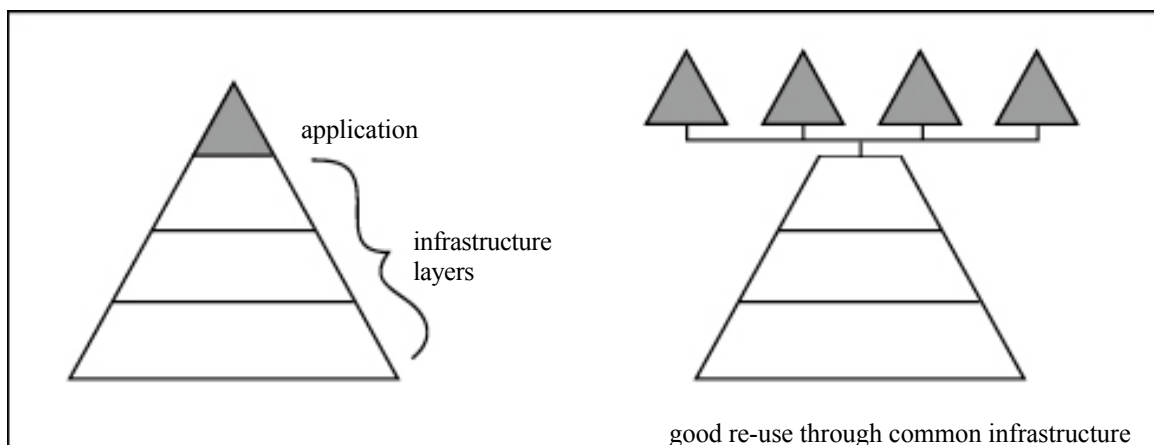


Figure 1 Interoperability through shared infrastructure (after MIT/OKI [7])

3 POOL, POND and SPLASH

Portals of Online Objects for Learning (POOL) was a previous (2000-2002) learning object repository project which emerged from Canada's TeleLearning NCE research program with partial support from the Canarie Inc. eLearning Program. While initial work was focused on building a single web portal, the authors' development team soon realized that a single portal would require an ongoing maintenance organization, and would have limited scalability. Recognizing that the best place for learning objects would be close to the user and close to the creator, POOL shifted to a federated peer-to-peer approach that could link robust community repositories (which we called PONDS) with individual desktop repositories (which we called SPLASHes). Without belabouring the water analogy, we foresaw a global POOL that embraced a variety of repository types interacting together to ensure a flow of learning objects between creators and learners, with each player gathering and holding those learning objects deemed to be of self-defined value. We recognized this architecture would need provision for archival and institutional storage in PONDS that could connect and interoperate with the peer-to-peer network. Figure 2 shows the conceptual architecture of this heterogeneous architecture. Note that we foresaw the necessary inclusion of "super nodes" that would route search request through Canada's high-speed Canarie research network – and thus a search request could be propagated in different parts of the country to attenuate the horizon effect often experienced in peer-to-peer networks. In implementation we discovered that any university node would automatically route search requests through the high-speed network, and thus this notion was redundant.

POOL also saw the development of an IMS compatible metadata application profile called CanCORE [10] that encouraged the uniform cataloguing of learning object records that were entered into the repository. Indeed, the SPLASH application consisted of the metadata editing engine and database, a flat file system for storage of the actual objects, a user interface for conducting simple textual searches and advanced metadata searches, and a peering protocol based on Sun's JXTA for the peering and file transfer mechanisms.

Although SPLASH achieved recognition as a successful prototype, we learned many valuable lessons that would shape our future endeavours in the follow-on project, eduSource Canada. The lessons learned from POOL were:

1. the expectations of expert communities such as librarians, instructional designers and course development professionals were much different than the academic instructors and learners. Indeed, although everyone assured us it was important to do advanced metadata searches, the reality of user observation studies is that most search queries involved open text searches across all fields – perhaps due to the familiarity of the Google (www.google.ca) interface.
2. while professional course developers enjoyed the ability to create templates for bulk treatment of learning objects and used many of the CanCORE metadata fields, the typical SPLASH users preferred less metadata, and a simpler interface. Indeed, we began to incorporate self-definable metadata templates because most user communities wanted to use their meta-data rather than the 83 fields prescribed by IMS/IEEE LOM. We foresaw the use of semantic web technologies to enable local metadata interoperating at the global level [3].
3. to meet the generic needs of various implementations, SPLASH needed to be configurable from a number of peering and database options. For example, we removed the JXTA peering software as it had difficulty tuning transient workstations and developed our own peering solution. Similarly SPLASH would offer a variety of database configurations using a small java data-base, HSQL, for individual workstations, and the larger MySQL for communities.
4. among the exciting results of POOL were our demonstrations of the use of SPLASH as an interoperability agent among diverse third party learning object repositories. Once repositories were linked to SPLASH, they were essentially linked to a growing learning object community. Interoperability thus became the focus of our contributions to eduSource Canada [6], a large follow-on project to build a learning objects infrastructure in Canada.

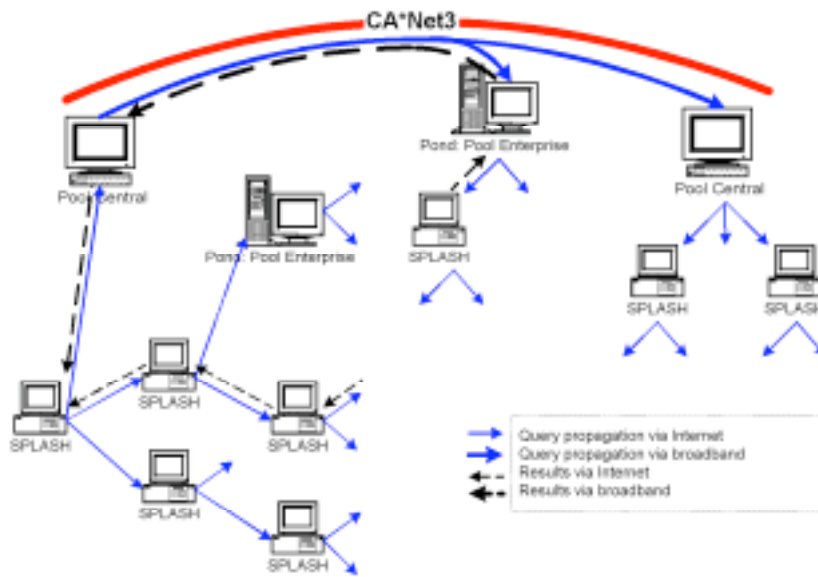


Figure 2. Propagation of peer-to-peer search requests in POOL, POND and SPLASH

4 Interoperability in eduSource Canada

A follow-on project to the Canarie e-Learning Program that sponsored POOL, eduSource Canada brings several university and industry partners together to build a framework for learning object repositories. While a broad description of eduSource is beyond the scope of this paper and may be found in [6] the basic notion is to enable learning objects stored in a wide variety of learning object repositories to be shared. In POOL interoperability was achieved by common use of the SPLASH application either as the repository, or as a custom connector to a community repository. In tackling the repository interoperability task in eduSource – we needed to deal with a wider variety of repository types running on a variety of platforms and with a varying degree of technical support. Additionally, we wanted to follow closely the IMS specification for Digital Repository Interoperability (IMS DRI) so that our work would have the best chance for uptake by repositories outside the eduSource community.

In defining the design and coding for what came to be known as the eduSource Communications Layer (ECL), a key goal was simplifying the task of connecting third party repositories. The ECL is a communications language in the web services based architecture as described by Paquette [8]. The ECL protocol implements a small number of functions (see Table 1) as defined in the IMS Digital Repository Interoperability Specification (IMS DRI)[4] such as search, gather, deliver, expose, etcetera. While these functions can be coded directly by a sophisticated web services programmer, we encourage them to use our middleware component, the “cobblestone connector”, which makes it possible for developers to implement the ECL without extensive coding.

The connector has a standard API to connect existing repositories to the eduSource network. The ECL protocol requires institutional repositories or tools to implement connector handlers only for those services they want to expose to others. This is a far simpler than implementing and deploying every service in each institution. The connector also facilitates synchronization of future protocol versions.

In addition to connecting individual repositories, the ECL also makes it possible to connect communities that are already networked using their own protocols. For example, we foresee connecting eduSource to the Australia’s Edna or the American NSDL. In these instances we favour the building of an ECL “gateway” between two communities: the ECL side is again a cobblestone connector but the other side defines a framework which allows developers to implement a chain of handlers to transform between ECL protocol and that of their community. This approach offers a clearly defined method for bridging

between two different protocols while partitioning the protocol transformation task into small manageable steps. In contrast to the simple connector which may only implement a few functions, a gateway enables the making all services available to all members of communities on either side of the transaction.

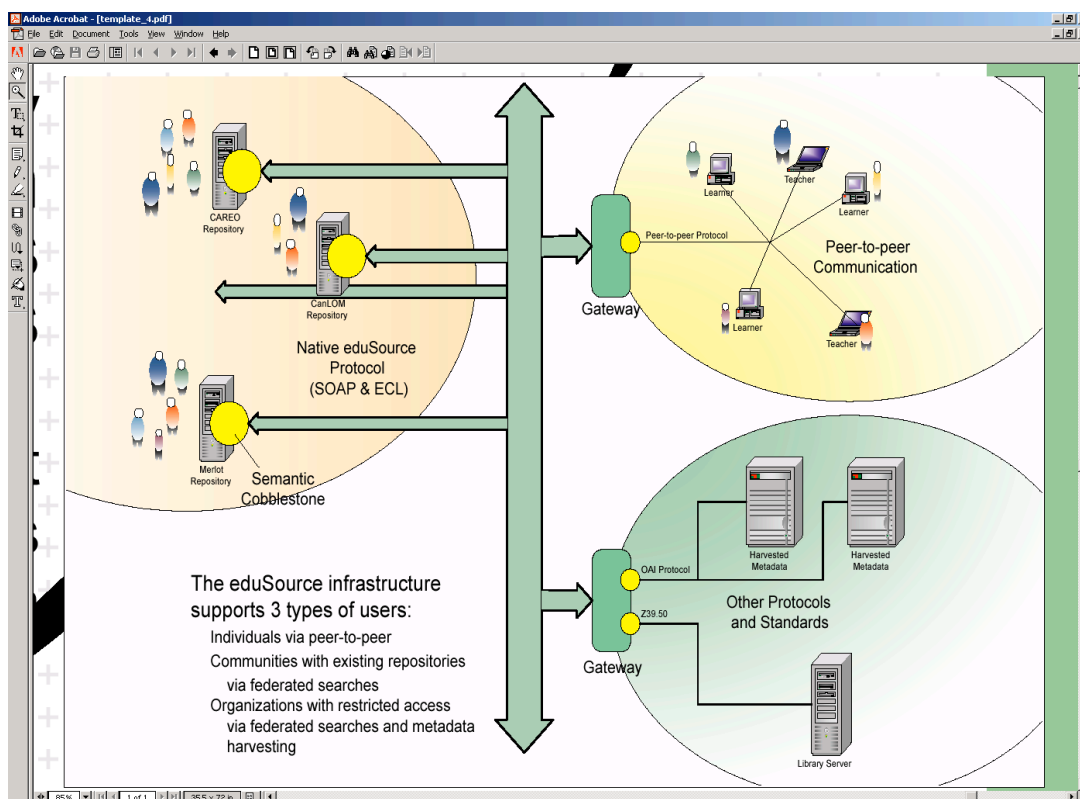


Figure 3. Sharing information via the eduSource Infrastructure.

5 Conclusions and New Directions

Effectively there are three ways in which those who want to connect to the eduSource community can do so. They can:

1. implement the ECL directly using SOAP.
This is rather complex and requires quite a lot of technical effort to comply with the protocol definitions and write code for each DRI function in building a connector and gateway for each repository (or repository network).
2. install and configure the connector and become direct participants in eduSource,
This is the preferred route as it requires a minimal understanding of the ECL protocol.
3. build a community gateway
This is advantageous when connecting a large community of users of a different protocol.

The ECL interoperability mechanisms offer the potential to connect a variety of learning object repositories and are an important test case for the IMS DRI. We expect the implementation of these components to be completed and all eduSource repositories should be connected by the end of 2003. As our goal is that other LOR communities will feel free to connect to this service, we have entered into Project LionShare with Pennsylvania State University and the MIT Open Knowledge Initiative to integrate our work in eduSource with the interoperability frameworks of OKI and to blend peer-to-peer technology with the InterNet 2 community's Shibboleth framework for security and authentication.

6 Acknowledgements

The authors acknowledge their valuable discussions of Jean-Francois Arcand in conceptualizing the ECL, and Ty Mey Eap in contributing to its development. The eduSource work at SFU Surrey is partly funded through the Canarie Inc. eLearning Program.

Table 1 ECL Services

ECL Service	Description
<u>Expose</u>	Expose is a SOAP service implemented on client computers. When asynchronous messaging is required, this service will be called by service providers to return the responses for search, gather, and alert.
<u>Gather</u>	Gather is a SOAP service implemented at service provider (server) computers. Repositories wanting to provide gather service must implement gather service handler.
<u>Gateway Gather</u>	Gateway gather is a type of gather service. It allows EduSource members to make gather requests to repositories external to eduSource. By default, the results are converted to IEEE LOM format or alternatively they remain in the format used in external repository Gateway gather service providers must specify a repository type and protocol (e.g. OAD).
<u>Search</u>	Search is a SOAP service implemented at service provider (server) computers. 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.
<u>Alert</u>	Alert is a SOAP service implemented client computers. IMS DRI recommends Alert for push gather. Repositories could send an alert to subscribing aggregators. Whenever repository has new metadata matching subscribe parameters, it sends an alert message to the subscribers.
<u>Submit</u>	Submit is a SOAP service implemented at service provider (server) computers. It is a function for moving an object (metadata and learning object) to a repository. Clients can make request to this service to determine the type of transfer protocol. The transfer protocol can be a successive SOAP request to upload the object or FTP transfer protocol.
<u>Store</u>	Store is a SOAP service implemented on client computers. When asynchronous messaging is required, this service is called by service providers to return the results of submit function.
<u>Request</u>	Request is a SOAP service implemented at service provider (server) computers. 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
<u>Deliver</u>	Deliver is a SOAP service implemented on client computers. When asynchronous messaging is required, this service will be called by service providers to return the chunked of results.

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] 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.

- [4] IMS DRI (2003) IMS Digital Repositories Interoperability - Core Functions Information Model, Version 1, Available: <http://www.imsglobal.org/digitalrepositories/index.cfm>
- [5] Koper R. Modeling units of study from a pedagogical perspective – The pedagogical metamodel behind EML, <http://www.eml.ou.nl/introduction/articles.htm>.
- [6] McGreal, R., Anderson, T., Friesen, N., Sosteric, M., Hewitt, K., Ring, J. MacLeod, D., Richards, G. Hatala, M. Calvert, T., Chiasson, M., Roberts, T., Carey, T., Harrigan, K., Paquette, G. and Downes, S. (2002) “eduSource: A pan–Canadian learning object repository” In Proceedings of the E-Learn 2002 Conference. Montreal: Association for the Advancement of Computing in Education.
- [7] MIT Open Knowledge Initiative Open Service Interface Definitions (Release 1.0rc5) 26 June 2003
- [8] Paquette, G. *et al.* The eduSource Suit of Tools: Use Cases Specification (v.06), June 7, 2003, Available: <http://www.edusource.ca>
- [9] Richards, G. (2002) The challenges of the learning objects paradigm. Canadian Journal of Learning and Technology, Special Issue on Learning Objects, **28** (3) Fall pp 3-9.
- [10] Richards, G., McGreal, R. Hatala, M., Friesen, N. (2003) “The evolution of Learning Object Repository Technologies: Portals for On-line Objects for Learning” Canadian Journal of Distance Education **17**(3).
- [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., Nejd, 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] W. Nejd, 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 Conference*, Honolulu, Hawaii, May, 2002.
- [14] Wiley, D. (ed.) The Instructional Use of Learning Objects, Association for Educational Communications and Technology, (2001) pp. 571–577.