PlanetDR: a multi-interoperable content repository

PUJOL AHULLÓ, JORDI; GARCÍA LÓPEZ, PEDRO
Department of Computer Science and Mathematics
Universitat Rovira i Virgili
Av. Països Catalan, 26, Tarragona
SPAIN

Abstract: - This paper discusses PlanetDR, whose architecture supports the interoperability of various educational digital repositories. It is based on the implementation of current open specifications for interoperability (such as IEEE LOM, IMS DRI and LORI SQI). This integration of different specifications should support better re-use of resources. Repository federation is also discussed as a mean for enhancing further this re-use.

Key-Words: - Educational repositories, Interoperability of learning objects, IMS Learning Design, LORI SQI, server federation

1 Introduction

Historically, the development of standards indicates that a particular process or technology is maturing and has achieved a degree of commercial success. Nevertheless, in learning settings, the adoption of standards involves a slow process for both educational institutions and commercial companies (standards tend to come first!). Although some learning standards are now sufficiently mature, such as LOM [1] and SCORM [2], their widespread adoption in institutions and software packages is still a difficult and slow process.

As regards learning repository interoperability standards, the problem is even stronger. Although a plethora of distributed content repositories have been implemented (for example Edutella [3], POND [4], ARIADNE [5]), the lack of interoperability among them hinders universal content aggregation in a single worldwide repository. As a consequence, there exist isolated content islands full of tagged LOM, Dublin Core and other kind of educational contents that are only reachable to a small or few communities.

There exist some standards that are focused on content repository interoperability. IMS Digital Repository Interoperability specification (IMS DRI) [6] is one of them. The IMS Digital Repository Interoperability Group provided a functional architecture and reference model for repository interoperability. Aiming at very broad application of the specification, the standard makes a recommendation only at a certain level leaving the resolution of more operational issues to the system implementers. This fuzzy specification leaves many open questions, and this mitigated against widespread adoption of a well-specified standard.

Another proposal is the Learning Object Resources Interoperability Framework (LORI) [7] which is part of the PROLEARN [8] project. This distinguishes between core services and application services, both of which require a common messaging infrastructure which enables repositories to interact (e.g. by means of XML-RPC, Java RMI, or WSDL/SOAP). In general, LORI follows a much simpler protocol than DRI, seeking to avoid the complexities of XQuery. This simplicity eases the implementation of LORI Simple Query Interface (LORI SQI) and thus lowers the burden of implementing Digital Repositories. On the other hand, it permits less flexible queries than DRI and thus limits content access and retrieval. LORI SQI is a widely accepted interoperability protocol in European settings in the projects ARIADNE and ELENA [9].

IMS Global Learning Consortium and other well-known entities are announcing that both library and educational environments would have to work in a wider interoperable context. In this way, initiatives like MIT's DSpace [10], that follows its own proposed OKI OSID standard, could become interoperable with educational repositories in a few time.

Reusability of the widely available educational contents can be raised by the union or federation of repository servers, like DSpace federation. This federation will be a useful tool for finding out assets on the nearest federated servers or, even more, on the whole network itself.

In conclusion, in the coming years, a key issue will be how LOM content islands, such as those

mentioned above, and other kind of repositories, like the library ones, can be integrated into a worldwide connected repository network. We propose that more scalable, robust and easily deployable technologies will be required to construct such large server federations. In this way, we present in this paper our open source content repository named PlanetDR. PlanetDR fully supports IEEE LOM, IMS DRI (ECL), IMS Content Packaging (IMS CP) and LORI SQI, and it also provides internationalization capabilities. We discuss how interoperability issues have been achieved and propose the p2pWeb server federation architecture. With this technique, which is a mixture of the web and p2p environments, each content repository itself can enter directly to the federation once it starts.
The rest of the paper follows this structure. In the following section we study in depth repositories interoperability; in section 3 we describe briefly IMS DRI (ECL); LORI SQI is detailed in section 4; we present PlanetDR content repository in section 5; section 6 concludes the paper.

2 Repositories interoperability

Repositories become interoperable when they can search for and exchange information, like courses, and use it correctly. Different communities have built interoperability standards to achieve this goal. These standards establish all the requirements for the interconnection and information exchange. Nevertheless, they may suffer lack of interoperability in the following ways:

1. Repositories follow different standards.
2. Repositories follow the same standards but with different query and result syntax.

The former is obvious, because they follow different standards and, therefore, they have no compatibility to the other ones. This is the case for current standards, including IMS DRI. IMS DRI sets specifically all web services via WSDL and their semantics, and that XQuery becomes the query syntax and IEEE LOM for results. But IMS DRI repositories can not interconnect with others like LORI SQI ones. In the latter, the standards themselves leave open, or simply make some recommendations for, the exact query syntax and their results format. This situation makes difficult real and successful repository interoperability. For example, IMS DRI sets specifically all web services and their semantics, and that XQuery becomes the query syntax and LOM for results. But IMS DRI repositories can not interconnect with others like LORI SQI ones. In the same way, the LORI SQI

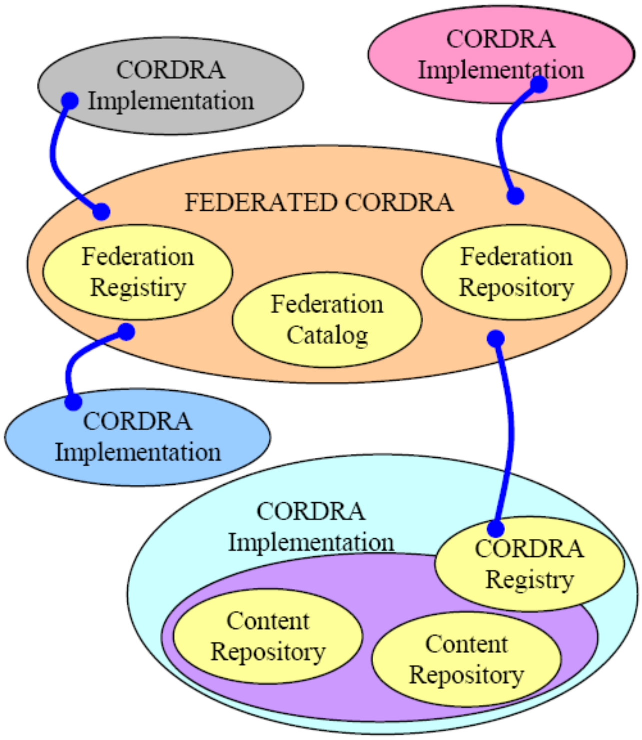

Figure 1. CORDRA federation

standard only specifies the query semantics. In this case, the query syntax could be VSQL (Very Simple Query Language), RDF and so forth, and the results format could be RDF, LOM, etc. Although two LORI SQI repositories would be interoperable, they cannot exchange resources if one only supports VSQL for query syntax, and the other one only supports RDF, for example.
In order to solve these problems, some initiatives have been presented in the last time for enabling effective repository interoperability. We think that more remarkable initiatives are CORDRA [11] and GLOBE [12]. They promote a federation of independent repositories. Federated repositories provide the following features:

- Discover content: Search for some kind of content, available on any repository from the federation, independently of its kind.
- Retrieve and share the content: According to the local repository authorization and rights. This allows private organizational resources not to be shared, but free educational assets be distributed in the abroad federation.
- Flexibility: Their implementations will be open and flexible, coexisting and interoperating with existing systems.

The natural extension of CORDRA is a multi-level hierarchy, where policies can limit for being a two-level system as shown in Fig. 1, for example. In this case, the top layer, namely *Feredared CORDRA*, provides global directory services. In the second layer there appear the whole content repositories, namely *CORDRA Implementations*. Here, *CORDRA Registry* records all content repositories with their full description, including their query and results syntaxes, and acts as a proxy for interconnecting the federation. The knowledge of these syntaxes for the queriers will enable a real interoperability between repositories. In CORDRA's specifications hints that developing a peer-to-peer (p2p) approach is also possible. In this case, another kind of operations must be treated, like replication, reliability, synchronization and scalability for a successful deployment.

On the other hand, GLOBE, in turn, focuses the interoperability between the well-known initiatives ARIADNE, Education.au [13], eduSource Canada [14], MERLOT [15] and NIME [16]. This effort is also directed to enable federated search across the boundaries of the different repositories. This seems to get a work-in-progress, because of the announcement in the ARIADNE web for first federated search with ARIADNE and MERLOT.

In the end, both CORDRA and GLOBE seem a good effort for a real interoperability, but, to the best of our knowledge, currently there not exists an available implementation for its use, and in GLOBE will be restricted to their repositories.

In the followings section we describe more deeply both IMS DRI (ECL) and LORI SQI standards, respectively, for presenting our content repository PlanetDR [17], showing details of their development and interoperability.

3 DRI and ECL

The purpose of the Digital Repositories Interoperabilty specification is to provide recommendations for interoperating between the most common repository functions. These recommendations should be implementable across services enabling them to present a common interface. DRI utilizes already defined schemas, such as IMS Meta-Data, mainly based on LOM and Content Packaging (CP) [18].

The DRI specification takes into consideration that a wide range of already implemented content formats, implemented systems, and established practices already exist in the area of digital repositories. Consequently, its recommendations lay out into two categories:

- Systems reflecting established practice (e.g. utilizing Z39.50 for repository interoperability).
- Systems that are able to implement the XQuery and SOAP-based recommendations.

Focusing on the second alternative, some core functions are defined as web services, which are exposed through the Internet, using SOAP, combined with WSDL (Web Services Description Language). This allows the content server to specify what services it provides, what the inputs/outputs of these services are, and how to encode/decode requests and responses exchanged between clients and servers. These core functions are described as follows:

Search/Expose: The search reference model defines searching through meta-data associated with content exposed by repositories. Searching is performed using the XQuery protocol over XML meta-data that follows the IMS Meta-Data Schema. XQuery has a well-defined grammar, and several commercial implementations are emerging from the community. Its strengths are query-by-example and structured searches of XML documents and repositories containing IMS meta-data.

Submit/Store: The submit/store functionality refers to the way an object is moved to a repository from a given network-accessible location, and how the object will then be represented inside that repository for access. The location from which an object is moved can be another repository, a learning management system, a developer's hard-drive, or any other networked location. It is anticipated that existing repository systems may already have established means for achieving Submit/Store functions (typically FTP). This specification provides no particular recommendations for legacy repository systems, but wishes to draw attention to the following weaknesses of FTP as a transport mechanism for learning objects or other assets: plain FTP provides no encryption capabilities, presents widely-recognized security flaws and does not provide means of confirming the successful delivery of assets from one networked location to another. In the case of more recently developed repositories that deal specifically with learning objects, this specification makes significant reference to the CP specification.

Request/Deliver: The request functional component allows users that have located a meta-data record via the Search function to access the content object or other resource described by this meta-data. Deliver refers to the response received from the repository which provides access to the resource.

Gather/Expose: The gather reference model defines repository-exposed meta-data requests, and meta-data aggregation for use in subsequent searches, or for creating a new meta-data repository. The aggregated repository becomes another entity available for Search/Expose functions. The gather component may interact with repositories either by actively asking meta-data from a repository, or by subscribing to a meta-data notification service. This notification service may be provided by the repository itself or by an external adapter that enables messaging between the repository and other users, thus following a push-based approach.

As mentioned above, one implementation of the DRI specification is ECL. This is part of the eduSource project, whose main aim is to create a network of linked and interoperable learning object repositories across Canada. A substantial part of the project has been the creation of communication protocols for sharing information as well as publishing the web services so anyone can tap their components into that pool of educational material and services.

Since the complexity of the ECL protocol might be detrimental to its adoption, an eduSource connector which implements the ECL protocol is provided. The connector provides a standard API to connect an existing repository to the eduSource network. The ECL protocol requires institution repositories or tools to implement connector handlers only for those services they want to expose to others, which is far 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 protocol, they only need to update the connector with a newer version. Changes in the ECL protocol are detected by the newer version of the connector and are dealt with automatically.

4 LORI SQI

The Simple Query Interface (SQI) is an open, collaborative effort, under the auspices of the CEN/ISSS Learning Technologies Workshop, with the collaboration of the excellence network Pro-Learn, to achieve interoperability between learning object repositories, with heterogeneous characteristics as assumption. Its proposal is to supply a very simple specification to become rapidly implemented and deployed. Their main properties are the presented below:

1. **Syntax-neutral**. It does not specify the query and result formats, due to the repositories heterogeneity.
2. **Stateful/stateless and synchronous/ asynchronous** services are both supported.

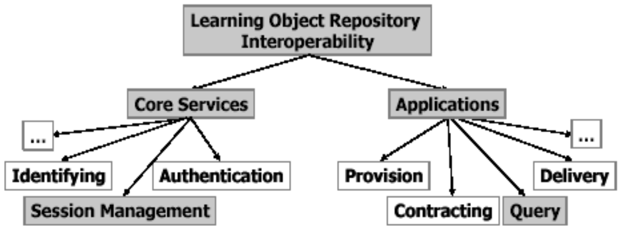

Figure 2. LORI SQI's application environment

3. **Session and request management** are separated, for allowing user authentication onto the system.

We show some details of these properties in the following lines.

Query and result syntax. For achieving a successful query/search system, the standard must specify the following aspects:

- Accepted attribute and expression sets.
- Its syntax and representation.
- How is represented the list of matching items.
- The metadata representation of metadata for items that match the query.

SQI does not define any of them, and leave to each repository set its own query and result syntax. For example, the result could be represented by LOM or RDF. Therefore, the interoperability between LORI SQI repositories is limited to those that follow the same syntaxes. Although SQI allows that each repository to adopt various syntaxes for both query and result syntax, this fact does not ensure a true interoperability between any two SQI repositories.

Stateful and stateless services. Stateful services are focused on recording the user session information. This session information helps to improve the system throughput, e.g. caching results of the last queries. In this case, new queries could match in the cache and a real query would not be performed. Stateless services do not save any information and, therefore, any new query is treated independently from the others. In conclusion, stateful services have more requirements, like memory capacity for saving the cache for all active sessions, and have more complexity on their implementation than stateless services.

Synchronous and asynchronous services. The main difference between both kinds of services is that in the former, the SQI client waits for the answer from the repository. In the latter, the SQI client identifies a mechanism for getting advises of results from the repository, but without an active wait for results. It is clear that synchronous services are simpler in terms of complexity and developing cost than asynchronous ones, but asynchronous services are more flexible for developing more complex services such as federated searches. In this kind of searches, the requested repository could forward the same query to other repositories in a certain fashion. Nevertheless, the behavior of both synchronous and asynchronous services depends on the own repository implementation.

Session management. The lifecycle of a SQI session is like in other kind of servers, such as FTP or HTTP servers. Before starting any communication with a SQI repository, it requires the establishment of a communication session. There exist two types of sessions, anonymous sessions and specific sessions, by proportioning user and password. After this step, the repository answers with a session identification, different of the other current sessions. This session is destroyed voluntarily by the SQI client or automatically by its expiration time. The SQI standard supposes the existence of secure mechanisms for providing protection on intrusion and interception, such as SSL services. For a simpler implementation, SQI also allows the use of a publicly available static session identifier. In this scenario, all queries are performed into the same session.

Request management. SQI uses the *Command-Query Separation Principle (CQS)* for enabling either commands or queries for each request. Commands only carry out specific actions and queries only return results to the SQI client. Therefore, no side effects can be performed when a query is executed and the other way around. This helps for an easier implementation and understanding of services.

5 PlanetDR content repository

The basic operation of a content repository is to provide the means for uploading resources, which are stored in a data warehouse. Later, these resources must be accessible to registered users (or publicly available) by allowing them to search contents by a broad variety of criteria.

Interoperability was a priority when designing our content repository. Fortunately, a Canadian network repository proposed a concrete instance of DRI, called Edusource Communication Language (ECL) [4]. PlanetDR has made a strong commitment to open standards and is the first open source learning repository that fully supports IEEE LOM, IMS DRI (ECL), IMS Content Packaging (IMS CP) and LORI SQI, with internationalization capabilities.

Their good qualities have started the interest to adopt PlanetDR as the content repository for various organizations, successfully achieved in MOREA [19] (Santiago de Compostela, Spain), and studied for universities and public organizations from other countries, such as Colombia, Ecuador and Mexico.

In the following sections we are going to describe our experiences on the development of both interoperability standards IMS DRI and LORI SQI, and our vision on repository federation.

5.1 IMS DRI development

We have implemented IMS DRI standard by means of the ECL initiative, described above. Therefore, PlanetDR supports XQuery for the query syntax, and LOM for the results syntax. The available web services include Search, Submit and Request services. PlanetDR includes several search types: the *quick search* function allows searching for keywords which match any of the metadata fields for a particular content; the *advanced search* function can be split into two additional types as well: *search by main metadata category*, where any LOM meta-data field can be specified, and the *accumulated search*, which allows searching for any field, linking together conditions of different LOM categories. In comparison to LORI SQI, the IMS DRI implementation has been long and quite tedious, due to the complexity of XQuery process and its mapping into the PlanetDR's database query.

One interesting feature of PlanetDR is the possibility of invoking any web services between PlanetDR repositories and content servers in the eduSource network. This is easily achieved because all of these servers follow the same ECL protocol. Nevertheless, the content server itself works as a standalone server, which makes it "unaware" of other content servers in the eduSource network.

5.2 LORI SQI development

We have developed LORI SQI with stateless and synchronous services, with a publicly available session identifier [20]. We have fixed VSQL as the query syntax and LOM as the result syntax. VSQL is a very simple XML format for querying by keywords. We can see an example in the following lines:

```
<simplequery>
       <term>keyword1</term>
       <term>keyword2</term>
       <term>keyword3</term>
       …
</simplequery>
```

In conclusion, this is an elegant and easy way for incorporating the LORI SQI standard to any content repository. As LOM results were already supported by PlanetDR, it did not add any other complexity. Therefore, its implementation has been quite easy

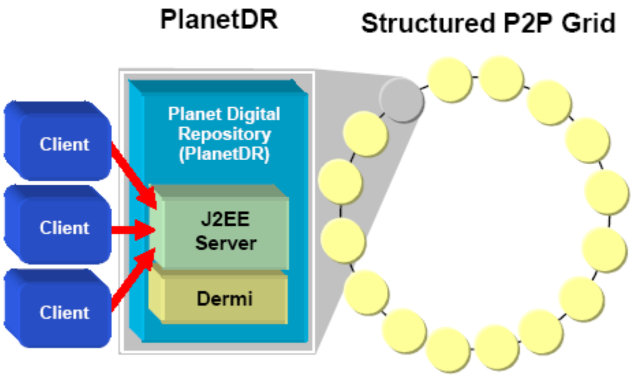

Figure 3. PlanetDR's Federation Architecture

and rapid, in comparison to ECL. The addition of more complex services, like session management, stateful and asynchronous services, has been planned for the future.

5.3 Towards a repository federation

In this section, we are going to describe our vision of an enhanced repository federation, following our experiences with PlanetDR. The goal of this federation will be always to improve the repository interoperability and others derived benefits, like automatic join of new servers into the federation, by means of the federation mechanisms themselves.

The past: a pure p2p repository federation. There is no way of easily knowing which other ECL content servers can interoperate with one content server. To solve this, the eduSource network linked servers by hand in a single central location. This approach clearly hindered the scalability of the federation if the number of servers increased.

To address this problem, we extended PlanetDR in later versions with a federation mode, using the federation architecture shown in Fig. 3. This mode supported plug & play decentralized management of PlanetDR compatible servers, thus guaranteeing worldwide scalability. New PlanetDR active instances in the network were automatically detected and inserted into each node's local list of available servers. Each PlanetDR node listened to different events which occurred (insert / remove), and this allowed each instance to maintain an updated list of available servers. Each server could join or leave the p2p federation, and get a list of all available educational repositories in the network. Thus the federated mode maintained "awareness" of both the identity of the nodes which made up the network, and also of the content which they held, so that directed searches could be sent to any of these nodes.

The overall PlanetDR federation architecture was scalable and could cope with a very large number of digital repositories, because it was based on the FreePastry [21] structured peer-to-peer overlay network. Specifically, PlanetDR was constructed onto a peer-to-peer middleware called DERMI, which was developed by the Planet project [22]. This provides a decentralized naming service and remote object notification mechanism. Thus, this technology provided a distributed and decentralized discovery mechanism for incoming and outgoing PlanetDR nodes, and they updated themselves the current existing nodes in a decentralized manner. For example, any incoming PlanetDR node was able to find all existing repositories in the system with a single lookup to the underlying DERMI.

Although this federation mode had good results and can seam a good effort, we currently are developing a new federation substrate based on crossing both p2p and Web paradigms: the p2pWeb.

The future: a p2pWeb repository federation. We propose the p2pWeb model, which offers decentralized solutions for service description, publication, discovery and availability, following the web services standards. Our p2pWeb model aims to bring all the benefits and unused resources of the edges of the Internet to the mature and standardized world of Web applications and services, although the web scenario is more complex and decoupled than in a traditional p2p scenario.

In this line, this infrastructure envisages a decentralized structured peer-to-peer network in which every peer hosts a web server. With this approach, existing web applications and services can make use of the resources of the network, but also obtain in a transparent way fault tolerance and load balancing services. Our concept goes further than other approaches like peer-to-peer web hosting (YouServ [23]) or peer-to-peer content distribution networks (CORAL [24]). Our p2pWeb approach offers true application deployment based on web standards on top of a peer-to-peer network.

The main difference is that with this new substrate, we will be able to offer innovative capabilities to the existing web development environment. We thus will provide a smooth transition from web applications and services to the decentralized peer2p model. Furthermore, our p2pWeb paradigm will be able to offer a number of key services like replicated data stores and databases, distributed hash tables, distributed naming systems, and scalable one-to-many communication channels.

Therefore, we aim to offer a new Service Oriented Architecture (SOA) for such p2pWeb scenario. We believe that all the features our p2pWeb model can offer will be of special interest for the creation of enhanced repository federations. Better integration with the repository, lightweight operation, federated searches, localized geographical searches, plug & play federation, data replication and publish/ subscribe mechanisms are some examples of the advantages of this new p2pWeb scenario.

6 Conclusions

In this paper we have discussed PlanetDR, which has been developed with the use of open specifications, such as IEEE LOM and IMS CP, enhancing the repository interoperability through the adoption of *various open interoperability standards*, namely IMS DRI (ECL) and LORI SQI. This adoption increases the reuse of educational content between different repository communities. Besides, we think that the simpler and easily developed a standard become, the more rapid its adoption can be. Thus, the clear and easy LORI SQI standard opens a door for a true interoperation beyond repositories, though their query and result syntaxes must be established publicly abroad beforehand.

In the other hand, reusability of educational content can be raised by a repository federation. We have shown our p2p and PlanetDR experience and announce our p2pWeb paradigm. The p2pWeb will be able to improve repository federation by offering enhanced services, like automatic addition of new repositories to the federation, federated searches through various repositories, data replication and a true scalability.

7 Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Technology through project PLANET (TIC-2003-09288-C02-00).

References:

[1] IEEE WG12, IEEE Standard for Learning Object Metadata. 2002. http://ltsc.ieee.org/wg12/

[2] ADL Technical Team, Sharable Content Object Reference Model (SCORM) 2004 2nd Edition. 2004, ADL. http://www.adlnet.org/downloads/70.cfm

[3] Nejdl, W., et al. EDUTELLA: A P2P Networking Infrastructure Based on RDF. *In Proceedings of WWW-2002*. 2002. Chiba.

[4] Hatala, M. and G. Richards. Global vs. Community Metadata Standards: Empowering Users for Knowledge Exchange. *In First International Semantic Web Conference*. 2002. Sardinia.

[5] The Ariadne Foundation, The two ARIADNE projects. 2004. http://www.ariadne-eu.org/en/foundation/history.html

[6] IMS Global Learning, IMS Digital Repositories Interoperability, Version 1. 2003. http://www.imsglobal.org/digitalrepositories/

[7] Simon, B., et al., Learning Object Repository Interoperability Framework. 2005. p. 8. http://www.prolearn-project.org/lori/,

[8] PROLEARN, Project Website. 2005. http://www.prolearn-project.org

[9]ELENA, Project Website. 2005. http://www.elena-project.org

[10] MIT Libraries and Hewlett-Packard Company, DSpace Federation. 2005. http://dspace.org/

[11] Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA), Project Website: http://cordra.net/

[12] Global Learning Objects Brokered Exchange (GLOBE), Project Website: http://taste.merlot.org/initiatives/globe.htm

[13] Education.au limited: http://www.educationau.edu.au/

[14] eduSource Canada: http://www.edusource.ca/

[15] Multimedia Educational Resource for Learning and Online Teaching (MERLOT), Project Website: http://www.merlot.org/

[16] National Institute of Multimedia Education (NIME), Project Website: http://www.nime.ac.jp/index-e.html

[17] PlanetDR, Project Website: http://planet.urv.cat/planetdr/

[18] IMS Global Learning, Content Packaging Specification, 1.1.4. 2004. http://www.imsglobal.org/content/packaging/index.html

[19] MOREA, Project Website: http://www.usc.es/morea/

[20] Public LORI SQI server list: http://ariadne.cs.kuleuven.be/SqiInterop/free/SQIImplementationsRegistry.jsp

[21] Rowstron, A. and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-Peer systems. *In International Conference on Distributed Systems Platforms (IFIP/ACM)*. 2001.

[22] Planet, Project Website: http://planet.urv.cat

[23] R. J. Bayardo Jr., R. Agrawal, D. Gruhl, A. Somani. YouServ: A Web-Hosting and Content Sharing Tool for the Masses. *In Proceedings of WWW-2002*, 2002

[24] Coral CDN, Project Website: http://www.coralcdn.org/