PlanetDR, a scalable architecture for federated repositories supporting IMS Learning Design

Josep Blat\textsuperscript{1}, David Griffiths\textsuperscript{1}, Toni Navarrete\textsuperscript{1}, José Luis Santos\textsuperscript{1}, Pedro García\textsuperscript{2}, Jordi Pujol\textsuperscript{2}

\textsuperscript{1}: Universitat Pompeu Fabra
Passeig de Circumval·lació, 8. 08003. Barcelona. Spain
{josep.blat, david.griffiths, toni.navarrete, joseluis.santos}@upf.edu
Phone : (+34) 93 5422237. Fax : (+34) 93 5422517

\textsuperscript{2}: Universitat Rovira i Virgili
Avinguda dels Països Catalans, 26. 43007. Tarragona. Spain
{pedro.garcia, jordi.pujol}@urv.net
Phone : (+34) 977 558510. Fax : (+34) 977 559710

Abstract

This paper discusses PlanetDR, whose architecture supports very large federated educational digital repositories. It is based on the implementation of current open specifications for interoperability (such as IEEE Learning Object Metadata and IMS Digital Repositories Interoperability, in its Edusource Communication Language version), and its integration with the workflow of eLearning production in the context of the Reload Learning Design editor. This integration should support better re-use of resources; some open problems for enhancing further this re-use are also discussed.

Keywords: Federated educational repositories, Interoperability of learning objects, IMS Learning Design, Pedagogy-aware services.

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 contents that are only reachable to small communities.

In theory, the unifying standard that should enable server interoperability is the IMS Digital Repository Interoperability specification (DRI) [6]. 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 mitigates against widespread adoption of a well-specified standard.

Fortunately, a Canadian network repository has proposed a concrete instance of DRI called Edusource Communication Language (ECL) [4]. PlanetDR has made a strong commitment to open standards and tools, supporting LOM and DRI, and is the
first Open Source Learning repository that fully supports ECL.

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 (XML-RPC, Java RMI, or WSDL/SOAP). In general, LORI follows a much simpler protocol than ECL and DRI, seeking to avoid the complexities of XQuery. This simplicity eases the implementation of LORI’s SQI (Simple Query Interface) 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’s SQI is a widely accepted interoperability protocol in European settings in the projects ARIADNE and ELENA [9].

Finally, MIT’s DSpace [10] is another Open Source Learning repository that includes federation capabilities. Although Dspace is not based on open standards, MIT has attracted a large number of Universities to the Dspace federation. The possibility exists that DSpace could reach critical mass and become a de facto standard in learning repositories. In conclusion, in the coming years a key issue will be how LOM content islands such as those mentioned above can be integrated into a worldwide connected repository network. This will be the case whether it is based on de facto standards such as DSpace, or well-specified protocols such as DRI or LORI SQI. We propose that more scalable and robust technologies will be required to construct such large server federations. The structured peer to peer architecture developed for PLANET which we present in this paper meets this need.

Looking at LOM repositories from another perspective we note that the retrieval of materials from educational repositories is an isolated task in the educational workflow. This isolation can hinder the re-use of educational resources, which we may take to be the goal of interoperable repositories. The retrieval functionality supported by PlanetDR becomes more fully meaningful if integrated into the process of creating learning activities, as discussed in the second part of this paper. To this end the Planet repository has been integrated with the Reload [11] editor, a reference Open Source tool for the creation of Learning Design Units of Learning. IMS Learning Design (LD) [12] is a recent specification allowing the representation of how multiple learners and teachers can work with resources in different activities. As a result of the integration work reported here it is possible to work with RELOAD, and without leaving the application query PLANET repositories, retrieve resources, and seamlessly incorporate them into “lessons”.

Finally, we discuss some open problems with the wider re-use of resources in this context. These include technical matters, such as the need for repositories to go beyond LOM based searching, and to provide full support for Learning Design based searches for resources, and also those of a more of social nature, such as supporting identification and re-use of the most successful resources.

The next section describes PlanetDR repository in detail, while the following section introduces LD and describes how Reload has been extended to deal with PlanetDR and the path to LD-aware repositories. The final section provides some conclusions.

2. 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 made accessible to registered users by allowing them to search contents by a broad variety of criteria.

When designing our content repository interoperability was a priority. We chose the Planet Digital Repository (PlanetDR) to implement the ECL protocol using web services. It also complies with the DRI interoperability specification, and both these specifications are described below.

3. DRI and ECL

The purpose of the Digital Repositories Interoperability 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) [13].

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, which PlanetDR is based on, 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. Although previous projects had informally created a distributed network that allowed the search and retrieval of educational objects between projects and organizations, there was no formal discussion of any best practice for the future. 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. Planet Digital Repository (PlanetDR)

Our educational content repository is called PlanetDR, and it is an implementation of the ECL protocol described above. The web services available include a search service, a submit service, and a request service. PlanetDR includes several search types: the quick search function allows searching for content keywords which match any of the meta-data fields for a particular content; the advanced search function can be split into two additional types as well: search by main meta-data 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.

One interesting feature of PlanetDR is the possibility of invoking any web services from other content servers in the eduSource network. This is easily achieved because all of these servers follow the same ECL protocol. In this case, what we call a federated search (a simulated gather service) can also be conducted by linking together request results coming from all active content servers in the eduSource network. Nevertheless, the content server itself works as a standalone server, which makes it “unaware” of other content servers in the eduSource network. There is no way of easily knowing which other ECL content servers can interoperate with it. To solve this, the EduSource network linked servers by hand in a single central location. This approach clearly hinders the scalability of the federation if the number of servers increases.

To address this problem we have extended PlanetDR with a federation mode, using the federation architecture shown in figure 1. This mode which supports plug & play decentralized management of PlanetDR compatible servers, thus guaranteeing worldwide scalability. New PlanetDR active instances in the network are automatically detected and inserted into each node’s local list of available servers. Each PlanetDR node listens to the different events which occur (insert / remove), and this allows each instance to maintain an updated list of available servers. Each server can join or leave the P2P federation of educational servers, and get a listing of all of them available in the network. Thus the federated mode maintains “awareness” of both the identity of the nodes which make up the network, and also of the content which they hold, so that directed searches can be sent to any of these nodes.

The overall PlanetDR federation architecture is scalable and can cope with a very large number of digital repositories because it builds on the FreePastry [14] structured peer-to-peer overlay network. Furthermore, PlanetDR is constructed on peer-to-peer middleware called DERMI, which was developed by the project [15]. This provides a decentralized naming service and remote object notification mechanism. This technology provides a distributed and decentralized discovery mechanism for incoming and outgoing PlanetDR nodes, and updates the current existing nodes in a decentralized manner. For example, any incoming PlanetDR node will be able to find all existing repositories in the system with a single lookup to the underlying DERMI.

![Figure 1. PlanetDR’s Federation Architecture](image-url)
5. Integration of PlanetDR and content creation

5.1 IMS Learning Design, Reload and the reuse of resources in context

The IMS Learning Design (LD) specification was produced to represent how multiple learners and teachers work with resources in different activities, a need not covered by SCORM, for instance. LD defines Units of Learning (UoLs) by representing how people carry out activities in an environment composed of learning resources and services. LD is a large and complex specification, and as a product of IMS it is by definition an interoperability specification, which does not constrain how eLearning applications should work, but only specifies an import and export format which they must be able to work with if they want to be LD compliant. At the risk of oversimplifying, a UoL can be seen as an interoperable lesson plan. In addition to being a file exchange format, however, LD is also an Educational Modelling Language, and a community of researchers working with LD has been established, building on the lead set out in the LD Best Practice guide [16] which proposes an eLearning methodology for the creation and use of UoLs. A set of applications has been developed to facilitate the creation and playing of UoLs using LD. All these aspects are reflected in the activities of the UNFOLD project [17] and interested readers can also find detailed information in [18].

Reload [11] is an Open Source editor of UoLs which sets out to be a reference implementation, that is to say that it implements the entire specification and provides a reference point for other developers who are in doubt as to how the specification should be interpreted. The very large number of downloads from the Reload website and the number of references to it in the published literature suggest that it has been successful in this. The equivalent Open Source implementation for runtime, the “reference player”, is CopperCore [19]. In its current version, Reload supports a single user on a single machine program, whose inputs, such as resources, have to be locally available, and whose output is a zip file packaged according to the CP specification.

While PlanetDR supports search and retrieval of resources based on LOM and DRI, a much more natural working context is to perform this task when an author is designing a UoL, and to be able to include the resources retrieved in the UoL. Thus, we have extended the Reload editor to allow to search and retrieve resources from the PlanetDR repository, and include them in the workflow of UoL production. We describe next how this is done and discuss the benefits.

5.2 Connecting Reload to resources stored on PlanetDR

A new window has been added to the Reload editor that enables the user to specify the fields for searching the resources in the repository. These fields are name, keywords and format. According to the values provided by the user, the tool builds an XQuery statement, which is sent to the repository and executed. The results of the query are presented to the user, who can select one or more resources from the list and download them in a zip file, in order to use them in the UoL that is being designed. The zip file is a requirement related to the CP specification and contains not only the resource but also the metadata file.

The tool interacts with two of the web services of PlanetDR implementing the ECL protocol, search and request. The first is called in order to send to the repository a query specified in XQuery. The query is executed on the LOM metadata files of the resources in the repository, and the service returns a string containing the list of resources that satisfy the query. Secondly the request service is called in order to download the resource.

Due to the very complex and manifold nature of eLearning, the daunting task of providing interoperability specification has been broken into pieces, such as LOM, DRI, CP, LD, and others not mentioned in this paper. This simplifies the task of specifications implementers, and makes compliance more practicable. For the user, however, this may create difficulties, as it can cause unitary tasks (such as preparing a course module) to be divided into seemingly unrelated parts. The user needs to have these specifications transparently integrated in a workflow, and indeed in many cases the user should not be aware of the various underlying
specifications. In our work we have integrated LD, DRI and LOM in the natural workflow for producing a UoL, showing that it is possible to make specifications more transparent to the user, and we believe this is a key step for usability.

Our work also indicates a possible path for performing this type of integration by re-using pieces of Open Source code, gluing them together through a web services approach. A general and open architecture for eLearning, which could be based on web services is discussed by Wilson in [20]; the SLeD project [21] has produced a prototype of such an architecture for LD allowing to plug new services (such as searching, blogging, …).

The CP perspective, which seems to have its origins in the ascendency of CD-ROMs, makes difficult to make full use of distributed resources (which might have their own rhythm of updating), and forces local downloading, re-packaging. A much more natural perspective in the context of the Web is to link directly to the (distributed) resource(s). It may be that this can be achieved by fuller use of CopperCore, which is an LD engine that supports services, rather than a player as such. This is one of the perspectives of further work, which will follow an approach similar to that of SLeD.

### 6. Further perspectives for future work

Another perspective of future work is related to fully utilising the potential of LD and taking into account social use. Current approaches only use LOM, and reuse is limited to resources. The LD specification should support reuse of pedagogy, services, etc. As pointed out in [22] there is a need for repositories to have LD awareness. An LD-aware repository could support searches for UoLs that have been used with a certain kind of content, retrieve fragments of UoLs, or provide metadata on the use of UoLs. It is reasonable to suppose that teachers will not simply identify and use UoLs on the basis of LOM, but will also, and perhaps more importantly, base their decisions on the practice of the mass of their peers, or of individuals who they respect. Consequently popularity is one of the reasons why resources or pedagogies will attract use by others. Moreover, for the identification and refinement of successful practice it is also necessary for the history of use to be represented. As much of this as possible should be done automatically, as it has been clear for some years that most users are highly resistant to adding metadata to resources [23], and the EduSplash Repository [4] takes into account these aspects, beyond the EduSource project. Another approach, more related to popularity, is currently being adopted by the Lionshare project [24]. Automatic analysis can show teachers which resources are popular in their area / age group / curriculum. Lionshare is using the Shibboleth system developed by Internet2 to create flexible trusted communities and in such a context it may be possible to identify the individual teachers who have been using the resources, enabling teachers to emulate the practice of their successful peers. We intend to investigate how the reworkings of UoLs are associated with the UoLs on which they have been based, to permit browsing up and down the hierarchies of parents and children.

An interesting and quite different approach to the support which repositories can provide users of UoLs is provided in [25]. She suggests that it may be possible to use Latent Semantic Analysis and indexing in order to find concepts and similarities of concepts within a corpus of UoLs. The degree to which this promising idea will be practicable is not yet clear, as stated in her conclusions on the approach, setting out a number of questions for further investigation: “Can it be used to classify designs as good as well as bad practices, for example when user data, such as success or failure rates, completion time, etc are added to the analysis, or even with human classification of the design? Are acts the smallest independent units in learning designs? Are the templates sufficient for practitioners to develop new courses?”

### 7. Conclusions

In this paper we have discussed PlanetDR, which as well as being based on open specifications, such as LOM and DRI, has an architecture which can support the very large federated repositories of the future.

We have also described and discussed the implementation of the integration of the searching and retrieving facilities of such a
tool into the actual workflow of eLearning production, which we deem as key for allowing the re-usability of resources, ultimate goal of repositories. We have discussed some open problems in this orientation of allowing the re-usability both from technical and social perspectives.

The modifications which we have made to the Reload LD Editor make a contribution towards expanding the functionality and improving the usability of repositories of eLearning Resources. It is, however, clear that this first step needs to be followed up by further work along the lines of future work which are indicated above.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Technology through project PLANET (TIC-2003-09288-C02-00), and by the European Commission through the Framework 6 IST Coordination Action UNFOLD.

References


