The OpenDock project: putting in place the infrastructure for sharing learning activities

Elferink, Raymond[1]: Griffiths, David[2]; Veenendaal, Edwin[3]

[1] & [3]: RayCom, Tiendstraat 41, Utrecht 3513EA Netherlands{raymond,edwin}@raycom.com

[2]: Universitat Pompeu Fabra, Passeig de Circumval·lació, 8, 08003 Barcelona david.griffiths@upf.edu

Abstract

The OpenDock project is introduced, and the approach to supporting the sharing of online educational resources outlined. The functional requirements for the OpenDocument repository are stated, and their significance discussed. The system which is currently being implemented is described. This is a light weight, open source, peer to peer system. The peer nodes are the users web presence, rather than their own computer, reducing the infrastructure requirements for small institutions and individuals. Support for IMS Learning Design is provided, with Content Packages being unzipped and stored as a hierarchy, and a parser provided to analyse the manifests and represent the UoLs. The effectiveness of the system in supporting sharing will be established in trials and demonstration activities in the OpenDock project.

Keywords: educational repositories, learning design, creative commons, peer to peer.

1. Introduction

The sharing of eLearning materials is not a simple matter, and for a review of the issues, see [1]. This is particularly true if the goal is to reuse learning activities and lesson plans in addition to learning resources. In an earlier paper [2] we argue that the use of Information Technology (IT) makes sharing harder than it was with paper based resources, and propose a framework within which the obstacles can be understood. The OpenDock project (www.opendockproject.org) seeks to stimulate the sharing of eLearning activities and resources in Vocational Education and Training (VET) by demonstrating how Units of Learning (UoLs) defined in IMS Learning Design can reuse learning resources and be shared between different institutions. This has involved the implementation of a repository designed to support the sharing of learning activities, and the first stage of this work is described in this paper.

There are many repositories which are designed to meet the needs of large educational institutions, which have their own powerful web servers, and technicians who can install and maintain software. Many smaller educational institutions do not have these resources, especially in the VET area directly addressed by the OpenDock project, funded by the European Commission through the Leonardo programme. Typically smaller institutions run their web through a web hosting service, and the task of establishing a permanent server to run a repository is a substantial one. This is still more true of individual teachers and learners who may want to run a repository. It is intended that any user who knows how to publish a web page will also be able to run the repository.

In this work we follow the analysis in [2] in identifying the need for revision of the functionality of eLearning resources repositories. It is not proposed that improved repositories will in themselves transform practice in sharing eLearning resources, but rather that they provide an essential element of enabling technology.

2. Functional requirements

In the light of this approach a set of functional requirements were developed, which are discussed below, with comments on their significance.

2.1 Support for Creative Commons Licenses

The copyright regime tends to restrict sharing of resources, not simply because resources protected by copyright cannot be

shared, but also because users are deterred from sharing resources because they fear the possibility that they might be infringing copyright. The Creative Commons licenses [3] provide a legal framework within which individuals and institutions can share and adapt educational materials without fear of losing control of their own work, or of infringing the copyright of others. OpenDock has chosen to work exclusively with materials licensed under the Creative Commons in order to maximise the potential reuse of the materials developed. This has significant implications for the repository to be developed, as it is not necessary to provide a complex rights management system. Support for applying Creative Commons licenses is included in the system.

2.2 Use of Peer to peer technology

Peer to peer (P2P) systems such as Kazaar, eMule, and Limewire are hugely popular among users. Among the reasons for their popularity are:

- They are easy to set up, with point and click installers.
- They do not have heavy hardware and software requirements
- They are effective since, as is argued by LionShare, [4]"A key trait of P2P is that it optimizes network usage by distributing it throughout the community of network users and thereby avoiding bottlenecks".
- They enable the user to set up server without requesting the permission of a system administrator, making it easy for teachers and learners to set up their own nodes, for example in project based learning.

The advantages of these systems have not been exploited as much as they might have been because of their association with illegal file sharing. This is issue is addressed by the application of Creative Commons licences to all materials published on the OpenDock system.

2.3 Support for IMS Learning Design

ELearning interoperability specifications provide formal and structured descriptions of resources. These can be leveraged by the repository, which can use the description to provide additional information to the user. This is particularly true of a complex specification such as IMS Learning Design.

2.4 Mechanism to comment on posted items

Formal metadata will be supported, but when deciding whether to use a resource on a repository feedback from other users is also a key factor, especially if they are made by known and respected peers, or by members of the same professional community. Support for this needs to be incorporated into the repository.

2.5 Authentication and groups

The Creative Commons license provides defence against copyright infringement prosecution by shifting the burden of proof onto the person claiming the license. If credible complaints are received that certain materials posted are protected by copyright, then it is important to be able to bar offending users from using the system. Consequently only authenticated users should be able to use the repository.

It is also important that users can belong to groups, so that a) comments on resources can be classified, and b) users can give access to their materials to specific groups or users.

Searching and downloading of published resources from the repository may be done by anonymous users

2.6 Services made available

A number of exciting new eLearning applications are appearing which use a service based architecture, and which can broadly be classified as Personal Learning Environments (PLEs). Oleg Liber describes PLEs as an alternative to institutional systems based on courses, and which *locate a large amount of VLE functionality with the learner either as a desktop application or an independently hosted portal. Institutions would still provide content via repositories, undertake assessment and so on, but learners would interact with these using their personal systems (Personal Learning Environment), comprising their preferred tools and ways of working.* [5] Emerging systems include Plex (University of Bolton), Hecate (Open University of the Netherlands) [6] and Elgg [7].

3. System requirements

The functional requirements and user profile informed the identification of the following system requirements:

- Minimal hardware requirements imposed on users.
- Easy install, without root access. Preferably, installation consists of uploading the files to the server and configuration through a web-interface.
- Entirely Open Source system.
- Extensible architecture.
- Support for RSS, and have a well documented API.
- Web-based interface.
- Distributed network of smaller peer servers.

4. Existing systems

A state of the art analysis showed that while there are many excellent existing systems, none of them entirely meet the needs of the OpenDock project. The principal problems are

- **High system demands**. Existing systems often make use of java or jxta, for example [8], and they often have quite significant system demands, which makes it very hard to run them on shared servers. Some systems need an Oracle database, for example Ariadne [9] which make them relatively expensive to run and maintain.
- **Complex installation procedures**, often consisting of multiple steps, including installation of extra libraries and software. Often root access is needed for some steps.
- Architectural issues. Even with the P2P systems the architectures often required a strong IT infrastructure.
  - Some systems, such as Ariadne use industrial strength central servers to store files and metadata, and federate searches over a network of institutional servers. These searches may be federated over a larger network by using a protocol such as the ECL (EduSource Communication Layer). This approach is well described in [10]
  - More typical peer to peer systems can be installed on any computer with an internet connection, for example LionShare [11]. These systems have to choose whether to store metadata and files on one machine only, or to propagate them over the network. The LionShare default solution is to keep the resources only on the node on which they were posted, and to propagate the metadata. This means that to ensure a reliable service the nodes have to be hosted on a machine which will be always on, and has a good internet connection (although mirrors can be set up). Propagation of the resources through the network, on the other hand, quickly creates very large amounts of data on each node.
- **Missing essential functionality.** Only some of the existing systems have support for Creative Commons licenses, while support for IMS Learning Design is very limited. The exception is Planet which is integrated with the Reload LD Editor, making it easy for authors of UoLs to find resources from the Planet network (see Blat et.al at this workshop). Indeed the Planet system is perhaps that which most closely approximates to that required by OpenDock, the principal difficulty being that, like LionShare, it needs to be installed on a reliable computer with a good internet connection. Another aspect of support for IMS Learning Design is that described in [12], which focuses on the use of ontologies to integrate learning designs with content.

5. Proposed solution

A small distributed web-based open source repository system called OpenDocument.net.

5.1 Architecture

As described in the previous section, prior P2P systems make demands on infrastructure which cannot be met by many of the target users of OpenDock. The problem is resolved by taking an approach which is similar to LionShare (for example), in that the metadata is propagated, but not the resources. However in our case the user's computer is not used as the peer, but rather a web presence. This may be an institutional server, but may more typically be rented web space. The only requirement is that the server be able to run PHP. In this network of rather independently functioning repositories, it does not harm the system if one particular repository fails to function. All repositories contain only the files

that were uploaded to them and on top of that the meta information about content at other nodes. In this way all the repositories are aware of the available content in the network, but there will be no integrity violations or racing conditions involved. If one node fails the system continues to function, although the other nodes will notice the failure and inform the user. This is similar to the Internet, which does not go down if a few servers malfunction.

Similarly this approach favours scalability (because only small metadata files are replicated over the network). Performance of any given node will depend on the choice of server made by the people who set it up. This enables institutions to make their own decisions about the quality of service which they wish to offer, the amount of files they want to share, and money which they want to invest. This decision can be delegated to the individual nodes because the performance of the network is independent of the performance of single nodes.

The solution consists of three parts:

- Repository Server
- Repository API
- Default user interfaces.

The logical model in which the data is stored consists of four levels:

- Network
- Repository
- Container
- Item

In this model a network consists of a number of repositories that replicate metadata and exchange information about their content. All content is stored in the Repository as items in a container. This could be a folder with images, a set of documents or a UoL with resources and a manifest file. Many repositories can only store one file at a time, either as separate entities, or as a zip file. In this approach complete directory structures can be stored, making it easier to handle HTML sites. Similarly a UoL uploaded as a zip file is expanded out to a directory structure, so that it can be searched and individual items returned in searches. It can be reconstituted as a zip file for delivery to the user. This architecture should provide a system which is easy to install and maintain, but which does not sacrifice scalability.

The name OpenDock.net recognises the relationship between this solution and the OpenDocument standard which is used in the OpenOffice toolset. In this standard the files are not considered as atomic units, but rather a collection of items. The way in which we organise the content in the repository is largely based on the ideas embodied in the standard, and the name OpenDocument.net has been chosen to make this association explicit.

5.2 Repository server

The server is a lightweight object oriented PHP application that can run on any web server which supports PHP and mySQL. Development and testing is being carried out with Apache. As stated above, it is intended that installation should be a straightforward task for anyone who knows how to publish a web page. The procedure consists simply of unpacking an archive (typically a zip-file) at the server site. Upgrading involves essentially the same process, the new files are unpacked and the upgrade is done.

Backing up is provided as a standard service by all hosting providers, but we also intend to build an administrative tool that backs up the whole repository in one or two files that are available for download to the administrator. This will also enable the state of the repository to be documented at any given moment. Restoring a consists of setting up the repository in an inactive state, and downloading or uploading the appropriate zip-files and unpacking them. If the database also needs to be restored too, a script is run to insert the meta data of the backup into a new database. This will be a simple two or three step protocol that is easy for the responsible administrator to perform. The system maintains data about the size of all the items and the repository as a whole. If the memory space limit of the server is reached, the user will be informed that uploading new content is not possible.The server has no built-in user interface. It consists of a core with a clearly defined API.

The server uses XML-RPC (www.xmlrpc.com) as a remote procedure protocol. This uses XML to encode its calls and uses HTTP as a transport mechanism. For added security the server makes use of HTTPS (HTTP encrypted by SSL or TLS protocols) for all external communications.

All items in the repository are stored in the file system where the server is installed. Metadata, user information and system information are stored in a database.

5.3 Repository interfaces

Initially the repository will come with a simple web-based interface application in PHP. This interface uses XML-RPC to communicate to the API of the Repository Server. The default web-interface will also be able to output RSS for aggregation to other (web)applications.

A clear API document will be available for developers willing to create their own interfaces to the Repository. Other potential interfaces could be connections from existing IMS Learning Design applications such as Reload or SLeD, or components for LMSs and CMSs such as Moodle, Joomla and Plone.

5.4 Authentication

Each repository instance maintains its own userbase. Users of a repository have access to their own repository and are able to add comments and metadata to other repositories in the network. The repository has an Access Control System (ACS) based on users, groups and rights.

5.5 Distribution

Information about the contents of a repository is replicated to other instances in the network. Replication will be done regularly or triggered by users with sufficient rights.

5.6 Support for open specifications

Currently planned IMS LD support for the OpenDocument Repository will leverage the storage of the zipped Content Package as one or more containers with the constituent files. This gives the user access to the individual resources and stimulates reuse. The Repository will have a built-in LD parser which can generate a simple preview/overview of the UoL, and make LD properties such as level, prerequisites, objectives etc. available for searches. The parser is a plug-in, and so it would be easy to later include support for other specifications, such as QTI or SCORM.

Support for LD authoring, such as that available for Planet, would also be desirable.

Items in the repository can be stored under Creative Commons licenses. The user uploading or creating a new item or container chooses the required license from the user interface. Creative Commons license information is also available for searches. The default interface will be able to generate RSS feeds from the Repository. Possible applications are: popular items, newest items, latest comments, etc.

5.7 Support for discussion

Users will be able to comment on items or containers, and their input will add to the metadata around the resource. Users also will be able to rate items.

6. Conclusion

The OpenDocument repository is currently being implemented, and trials are scheduled to start with end users in October of 2006. Its light infrastructure demands, simplicity, support for IMS Learning Design and Creative Commons, and service orientation are a unique combination of features which will hopefully make it possible for small institutions, and individual teachers to establish reliable educational repositories without major investments of time or resources, and to establish effective practice in collaboration. This will be tested in the OpenDock project during the second half of 2006 and 2007. Those beyond the project who would like to participate in this process are encouraged to contact the project by writing to opendock.info@upf.edu

References

1. Littlejohn, A., ed. *Reusing Online Resources: A Sustainable Approach to E-Learning*. Open and Flexible Learning Series. 2003, Kogan Page Ltd.

2. Griffiths, D. *Print to Pixels: the Implications for the Development of Learning Resources*. in I2LOR 2005, second annual e-learning conference on Intelligent Interactive Learning Object Repositories. 2005. Vancouver: November 14-18.

3. Creative Commons, "Some Rights Reserved": Building a Layer of Reasonable Copyright. http://creativecommons.org/about/history

4. LionShare, *FAQ*.

2005.http://lionshare.its.psu.edu/docs/build/site/users/faq.pdf

5. Liber, O., *The personal learning environments blog*. 2005, Bolton University.

http://www.cetis.ac.uk/members/ple/resources/ple_summary

6. *Hecate Project,* Sourceforge download. http://sourceforge.net/projects/hecate

7. *Elgg project website*.http://elgg.net/

8. Nejdl, W., B. Wolf, and C. Qu. *EDUTELLA: A P2P Networking Infrastructure Based on RDF*. in WWW2002. 2002. Honolulu.

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

10. *Richards, G., et al., The Evolution of Learning Object Repository Technologies*: Portals for On-line Objects for Learning. Journal of Distance Education/Revue de l'Éducación a Distance. TL•NCE Special Edition/Édition Spéciale RCE•TA,, 2002. 17(3): p. 67-79.

11. *OKI Case Study, The Repository OSID and LionShare: Leveraging standards to provide broad markets for content (Draft)*. 2005, Massachusetts Institute of Technology.http://www.okiproject.org/documents/CaseStudy-Lionshare.pdf

12. Knight, C., D. Gaševic, and G. Richards, *Ontologies to integrate learning design and learning content*. Journal of Interactive Media in Education (Advances in Learning Design. Special Issue, eds. Colin Tattersall, Rob Koper), 2005(07).