Finding the Right Tool for the Community: Bringing a Wiki-Type Editor to the World of Reusable Learning Objects

Chu Wang[2], Hugh C Davis[2], Kate Dickens[1], Gary Wills[2]
[1]*eLanguages, University of Southampton, UK*
[2]*School of Electronics and Computer Science, University of Southampton, UK*
[1]*kate.dickens@soton.ac.uk,* [2]*{cw2, hcd, gbw}@ecs.soton.ac.uk*

Abstract

In this paper we present a new approach to enabling pedagogically sound reuse and re-purposing of online learning objects in a community of practice. The lack of specific software for non-technical users inspired the development of an innovative toolkit. The toolkit allows users to edit learning objects collaboratively in a Wiki-type authoring environment and to add and modify context-rich metadata which greatly assists in the efficient retrieval of learning objects. The focus of this paper is on the development of the Wiki-type editor which provides an effective way of re-purposing learning objects in disparate communities.

1. Introduction

More and more educators realize the benefits of using e-learning materials as a practicable alternative to traditional instructor-led courses, hence the usage of learning objects has become prevalent in the education world. Polsani [11] has defined a learning object as "*an independent and self-standing unit of learning content that is predisposed to reuse in multiple instructional contexts*". This definition suggests several functional requirements which are essential for creating sensible learning objects. For example, they need to be stand-alone, reusable, tagged with metadata, and be able to be aggregated. Of particular importance is that sharing and reusing e-learning materials may lead to an improved quality of teaching, the sharing of good practice, greater consistency and an enhanced sense of community [1].

Different aspects of learning objects need to be considered in order to promote reusability. One important factor is metadata which facilitates the identification, search and retrieval of learning objects and can be extended to represent different instructional contexts. IMS/GLC [12] has provided specifications and guidelines for metadata standards and extension rules. Context-rich metadata [7], in other words contextual metadata, plays an important role in understanding the reusability of a learning resource. Here context has been defined as "*A set of circumstances in which a learning object is used or may be used*" [2]. Particularly the context refers to the teaching and learning circumstances. Contextual metadata can represent information about the intended target audience, the purpose or instructional methods, pedagogic approach being used and so on. Learning objects with such information attached can then be identified by their context and re-purposed to suit different needs.

It is possible to reuse learning objects in a variety of different ways. Educators can take an existing learning object or generic template and produce a different version of the same material; or they can constrain the size of the material, for example if it were to be viewed on different devices such as PDAs or mobile phones. In most scenarios, they need to tailor the existing objects to target specific learner groups or different instructional contexts.

Inevitably it is challenging to reuse learning objects in differing contexts [5] especially in relation to the issues about extensions on relevant standards, exchange formats for contextualisation of resources, creation of tools for development of contextualized learning resources. It is the authors' intention to address these issues by developing novel toolkits and using the existing repository of learning objects and established community as a test bed.

The remainder of this paper is organised as follows: Section 2 suggests our approach to reusing learning objects and proposed toolkits. In Section 3, we discuss the need for a Wiki-type editor and illustrate how we designed it by replicating traditional Wiki mechanisms; while section 4 describes the implementation of such a tool. Finally the authors report feedback from an initial evaluation, and conclude with a summary of the paper and explain the future work in our project.

2. Our Approach for Reusing and Re-purposing – The MURLLO Project

The last few years has seen a debate arise between the approaches to designing learning objects in contextualized

and "de-contextualized" scenarios [2,3,4]. Although learning objects are widely developed as free from the context of teaching and learning to facilitate interoperability, the research carried out within the eLanguages group at Southampton[1] found that contextual metadata facilitates greater scope for reuse. Reuse is significantly improved through the inclusion of additional metadata, which describes the pedagogic nature of a learning object. Indeed we are not alone, several other projects discovered the benefits of using contextual metadata, such as RAFT [5], ProLearn Query Language (PLQL)[2], and the digital library DocSouth [6], and our work draws on the experiences gained in these projects.

The MURLLO (Management, Use and Re-purposing of Language Learning Objects) project run by the eLanguages group aims to tackle concerns surrounding the effective re-purposing of Reusable Learning Objects (RLOs). We proposed a system with three components/tools to deal with these concerns

- ***The Wiki-Type Content Editor*** allows re-purposing of content without the need for extra authoring tools and the storage of this content and its revisions.
- ***The Metadata Facilitator*** acts as a teacher-friendly interface to allow the application of customized context-rich metadata to RLOs and facilitate storage of resultant content packages in open and closed repositories.
- ***The Discovery Agent*** facilitates resource discovery by displaying contextual metadata and by allowing selection and export of collections of RLOs.

The Wiki style content editor is a critical component which enables a non-technical user to adapt learning objects. It directly affects how well users can achieve their repurposing targets. We will focus on the Wiki-type tool in this paper.

3. Wiki-Type Editor Requirement and Design

As pointed out in [9], the ease of collaboration in a Wiki can make it a powerful tool for project management and collaborative writing. In our case, a Wiki-type authoring tool provides the necessary capability for the adaptation and repurposing of learning objects.

3.1. The Community Need for Wiki-Type Editor

There is research [3, 10] which focuses on the design pattern or best practice for learning objects intended for re-use in other contexts. However, we have a different perspective. Our previous research [7] has shown that it is easy to reuse pedagogic assets such as standalone audio files. However, it might be difficult for a non-technical educator to adapt HTML content within online learning objects for different target audiences. The need for an easy-to-use authoring tool has been stressed many times by our community of users.

Our community comprises practitioners from four regional universities in UK Higher Education (HE) sector. They are mostly language teachers but also some learning technologists and researchers. Most of the practitioners did not have any previous experience creating learning objects. Even they have started to share and adapt learning materials stored in the community repository in the earlier project L2O [7], they still need technical support because of their limited knowledge of web related technologies. Therefore demands for authoring and adapting online learning objects according to instructional contexts have emerged, which fits in with traditional practitioners' experience in adapting materials to suit different student groups (this finding was also supported by the attendees of an extended workshop[3]). Furthermore, we discovered that, in a collaborative online environment, expertise can be shared, exchanged and viewed by the community to achieve adapting and authoring goals. To rectify the current lack of relevant tools, the MURLLO project has proposed a Wiki-type editor to fill the gap. A Wiki is a type of Web server that allows any reader of its pages to edit those pages, or create new ones using simple web forms [8]. By using a Wiki-type editor, users' demands for collaborative content authoring can be achieved. And keeping different versions of adapted learning objects can help the community to track and view changes if needed.

3.2. Comparison of Potential Toolkits

Before designing our Wiki-type editor, we investigated whether any existing toolkits could be used with minor modifications. Authoring tools specifically for learning objects were the first options we looked at. Although some of them can generate impressive interactive learning content such as Xerte[4], or allow users to produce component-based learning objects without proficient knowledge of HTML and XML, like eXe - eLearning XHTML editor[5], unfortunately they are mostly offline editors and would be more suitable for creating learning objects from scratch rather than modifying existing objects. This investigation led us to believe that a Wiki-style editor would be the most attractive solution as an online collaborative authoring environment which best suits our needs. With this decision, it was a natural inclination for our team to consider using or adapting an existing Wiki system. A number of software/systems, including Wiki toolkits and document/content management systems, were explored in order to choose the most appropriate solution.

[1] http://www.elanguages.ac.uk/research

[2] http://ariadne.cs.kuleuven.be/lomi/index.php/QueryLanguages

[3] e-Learning conference, University of Southampton, Feb 2007

[4] http://www.nottingham.ac.uk/~cczjrt/Editor/

[5] http://exelearning.org/

Table 1: Comparison of existing tools (×not available; ○ partially available; ●fully available)

Tools / Required features	MediaWiki	DokuWiki	TWiki	Plone	Silva
Treat LO as a whole package/folder	×	×	○	●	●
Display LO (HTML pages) as it is (how many HTML tags can be recognized)	○ recognize some tags	×	○ large portion of tags but not all	○ apart from styles	○ apart from styles
HTML editing (WYSIWYG)	○ plugin	○ plugin	●	●	●
Import HTML file or zip file (content package)	×	×	○	●	○ not stable
Export HTML or zip file	○ simple html	○ simple html	○	○	○
Versioning	●	●	●	○ add-on products	○ after publishing
Versioning for media files	●	×	●	○ add-ons	○
Allow embedded media (Flash, video)	○	○	○	●	●
Image editing (or other media files editing)	×	×	○	○ from html editor	○ from html editor

Wiki Software

We looked into a number of different Wiki environments, including the one behind the popular Wikipedia (Mediawiki) [13], a simple and easy-to-use Wiki (DokuWiki) [14], a Zope based Wiki (Zwiki) [15] and many more. Table 1 shows the comparison of some of these Wikis based on the supporting technologies and the features they provide. Some of the features important to our application include user-friendly WYSIWYG toolbars, versioning capability, and the ability to handle embedded media (as many of our learning objects use flash applications).

Quite a few of them fitted in well with the above criteria. However, we were confronted with difficulties when we tried to import our existing learning objects into any of the aforementioned wiki environments. Our learning objects are web pages which are composed of standard HTML markup (with links to images, multimedia and embedded Shockwave Flash™ objects), whereas Wikis are created using their own markup language which can be totally different from HTML. Therefore a learning object always needs to be converted into the format used by the Wiki before it can be used in the Wiki editor. However it would still need to be converted back to HTML before the learning object could be exported from the system. The underlying problem was that the "round-trip" conversion between HTML and Wiki markups would cause a loss of information from the HTML web pages, especially embedded object tags and snippets defining page styles. If this information was not retained, both their look-and-feel and their functionality would be affected. This would have a serious impact on users since they would be unable to re-purpose materials if they could not see how the original learning objects had appeared. One possible solution would be to keep track of relevant data in the original HTML and then put it back when a user tries to view or export a learning object. However it would be error-prone dealing with HTML in this way as some information might not be resumed or might cause conflict to users' modifications which are not predictable.

Furthermore, most Wikis do not have a function to import HTML files or zipped packages. It is not appealing for a user to break down an existing package and upload components separately without the potential risk of changing the original structure of a package. For all these reasons we concluded that existing Wiki software was an inappropriate solution for implementing our Wiki-type editor.

Document Management Systems

Another type of software considered was Document Management Systems (DMSs), which are systems used to track and store electronic documents and/or images of paper documents. In particular, we investigated TWiki [16], another fully functioning Wiki system and a document management system. TWiki has a built-in WYSIWYG editor which makes it easy to edit HTML pages directly in a Word style environment. Unfortunately we experienced a similar problem of conversion between HTML and Wiki markups, as TWiki does not handle all HTML tags and it stores documents in Wiki format so post-editing reinsertion would also be necessary. A particular problem we encountered was that it can not display flash objects properly in its editor. The built-in conversion engine could deal with simple html documents

produced from its editor but was not ideal for our learning objects with media components embedded. As a result, DMSs offers us some advantages in terms of managing learning objects as packages but the same problem of conversion of HTML to Wiki formats still remains.

Content Management Systems

Content Management System (CMSs) shares many of the same concepts as those of DMSs. They offer a few more useful features compared to a DMS. For example, they usually have a built-in content importing feature so that different types of files in a package can be uploaded to a CMS system in one upload. We have investigated two Zope based systems, Silva [17] and Plone [18]. They both have built-in HTML editors, but certain HTML tags especially styles still get removed once uploaded. In addition, version control in these systems is not ideal for our case. They were either designed for publishing, like Silva, whose version information is visible only when a document is published; or like Plone, which lacked mature versioning functions at the time our editor was designed. Some add-on products were available but they were not suitable for versioning of content packages.

3.3. HTML Editor Integration

We compared several off-the-shelf Wikis and CMS/DMS toolkits based on a variety of characteristics crucial to our case, and we discovered that none of these tools particularly fitted our requirements (Table 1). Although CMS tools appeared to have more advantages, they were lacking in some key features such as versioning and keeping styles. Hence we came to the conclusion that it would be more efficient to design our own system by integrating an HTML editor and develop other essential functions like versioning and import/export facilities around it.

3.4. Versioning for Learning Objects

It is an important matter to decide how to define and store versions of learning objects being edited in the toolkit as this facilitates adaptation and reusability, while allowing users to be aware of all the changes made and act as protection against potential mistakes that might occur during the editing process.

At this stage the learning object editor was designed to deal with the learning material content which includes possible manipulation of media files (a facility to edit metadata will be incorporated later in *Metadata Facilitator*). It is necessary to consider versioning learning objects as whole packages including media attachments and metadata, but not as in traditional Wikis, where versions are handled at page level (single document). Hence package level versioning was adopted to accommodate the nature of learning objects and future extension on the toolkit to edit metadata and other attachments in content packages.

Users could re-purpose learning objects from different perspectives. The changes can be minor, for instance changing spellings or replacing words, or major, such as restructuring for different pedagogical designs. In the latter case metadata is likely to be modified. Discussions took place to decide whether to make new versions for each minor modification or only associate each major change with a new version. After careful consideration, we agreed that versions should always be kept instead of overwriting existing copies no matter how many changes have been made. As some modifications towards original learning objects could be controversial or subjective and may only reflect personal opinions, it would be wise to keep a full version history with rollback facility so that a faulty change can be reversed.

4. Wiki-Type Editor Implementation

According to the design, we chose a lightweight but powerful HTML editor, FCKeditor[6], to integrate a *Wiki-type* online authoring workspace for our community. We labelled it "Wiki-type" here since we have replicated a Wiki's collaborative authoring characteristics around a regular HTML editor. It is intended for peer-editing of existing learning objects but is not a "Wiki" in the more general sense of the term. As it was designed for teachers without web expertise, we have made the interface as intuitive as possible. The choice of HTML editor made the implementation process relatively smooth and with certain customisations it was turned into an ideal authoring tool for learning objects. A learning object can be uploaded as a standard IMS content package, then the system handles de-packaging which separates the learning object content from metadata and other documents. The only step users need to take is to locate the learning object from an index page and click the title link which will lead to the learning object being displayed in their browser. The look-and-feel of a learning object is kept consistent even in the editing mode so that novice users could edit it without being confused by the underlying HTML syntax and they can edit the content as it appears to their students. The main features of the system are listed below:

- A WYSIWYG editing environment for content authoring suitable for non-technical users;
- A versioning system that creates a new version after a learning object is modified and stores all previous versions of learning objects;
- Rollback and viewing and of earlier versions;
- An edit summary shows the nature of changes, to be used as a metadata field defining the relations between different versions of the same learning object;

[6] http://www.fckeditor.net/

- The ability to handle embedded media objects and allow relatively easy editing of these components;
- Import/export facilities for content packages, which will be linked with the metadata facilitator and discovery agent at a later stage.

5. User Evaluation

In order to check and validate whether the Wiki-type authoring tool is appropriate and useful for teachers, we undertook a qualitative evaluation during a workshop given to language teachers and learning technologists from a range of HE institutions within the UK. Attendees were asked to explore the tool by using it to adapt learning objects. They were given a free choice as to the method they employed in order to evaluate the usability and effectiveness of the tool. Twenty one questionnaires were returned at the end of the session and during the session the development team observed the way the attendees interacted with the system. The general observation made was that most users managed to edit learning objects within a few minutes.

The feedback we received from the questionnaires was positive; 86% of the users had a very good impression of the tool; 90% found it easy to navigate and obtain the information they wanted; 75% regarded it an effective tool for adapting learning objects; and 70% considered it easy to become familiar with the tool. This indicated the authoring tool is useful in repurposing learning objects and has the potential to be adopted in the current community. Many of the users could see how this tool could be used beyond its original scope. For example by including templates we could allow users to develop new and innovative learning objects from complicated components, something many of them find hard to do today.

6. Conclusion and Future Work

In this paper we have described our efforts to encourage and facilitate the reuse and re-purposing of online learning objects within a community of practice with little or no technical expertise. We have reasoned why a Wiki-type authoring system could help such a community to adapt learning objects collaboratively via the Internet. We have explained the rationale for the design of an authoring tool which reproduces the collaborative editing mechanisms of traditional Wikis for use by the community. The integration of a mature HTML editor has proved to be very effective in the case of adapting online learning materials for non-specialist users. The tool has been evaluated and received positive reactions within a group of community members.

We identified from our earlier work that the addition of extended forms of contextual metadata embedded within the learning materials would provide a wealth of new information with which to enhance the discovery and reusability of learning objects. However adding or modifying metadata could be very difficult for non-technical users without support. This leads to the *Metadata Facilitator* being developed which will bridge this gap. Finally we are planning a discovery agent to facilitate resource discovery and selection. All three tools will be integrated to form a single toolkit to help the management and reuse of learning objects.

7. References

[1] Barker, E., James, H., Knight, G., Long-Term Retention and Reuse of E-Learning Objects and Materials, report commissioned by the JISC, 2004.

[2] Johnston, P, It's all in the context. eFoundations metadata, middleware, e-learning, 2006. http://efoundations.typepad.com/efoundations/2006/12/its_all_in_the_.html. Last accessed 23 Feb 2007.

[3] Jones, R. Designing Adaptable Learning Resources with Learning Object Patterns, Journal of Digital Information Volume 6 Issue 1, Article No. 305, 2004-12-10.

[4] Jeffery, A. Context, metadata and e-learning: a literature review, project report 2006. http://dash.ecs.soton.ac.uk/dokuwiki/doku.php?id=documents:context_metadata_and_e-learning_a_literature_review. Last accessed 23 Feb 2007.

[5] Specht, M., Kravcik, M. Authoring of learning objects in context. *International Journal on E-Learning,* 5**,** 25-33. 2006.

[6] Pattuelli, M. C. Context for content: Shaping learning objects and modeling a domain ontology. IN Blandford, A. & Gow, J. (Eds.) Digital Libraries in the Context of Users' Broader Activities. First International Workshop, DL-Cuba 2006. *Chapel Hill*, NC USA, DL-Cuba.

[7] JISC funded project Sharing Language Learning Objects (L2O). available at http:// www.elanguages.ac.uk/sharing.

[8] Leuf B., Cunningham W. The Wiki way, quick collaboration on the Web, 2001, Boston:Addison-Wesley.

[9] Wei, C., Maust, B., Barrick, J., Cuddihy, E., Spyridakis, JH., Wikis for Supporting Distributed Collaborative Writing, the Proceedings of the Society for Technical Communication 52nd Annual Conference, Seattle, WA, May 8-11, 2005.

[10] Kravcik, M., Specht, M. Authoring Adaptive Courses – ALE Approach, Fraunhofer FIT. Sankt Augustin. Germany.

[11] Polsani, P.R., Use and Abuse of Reusable Learning Objects, Journal of Digital Information, Volume 3 Issue 4, Article No. 164, 2003-02-19.

[12] IMS Global Learning Consortium, documents available at http://www.imsglobal.org/

[13] MediaWiki. http://www.mediawiki.org/wiki/MediaWiki

[14] DokuWiki. http://wiki.splitbrain.org/wiki:dokuwiki

[15] ZWiki. http://www.zwiki.org

[16] TWiki. http://twiki.org/

[17] Silva. http://www.infrae.com/products/silva

[18] Plone. http://plone.org/