An Information Architecture to Support Dynamic Composition of Interactive Lessons and Reuse of Learning Objects

Melanie Kellar, Henry Stern, Carolyn Watters, Michael Shepherd
Faculty of Computer Science
Dalhousie University
Halifax, N.S., Canada B3H 1W5
{melanie | stern | watters| shepherd @ cs.dal.ca}
Contact author: shepherd

Abstract

In this paper we describe an information architecture that supports the dynamic composition of Web based lessons based on a database of fine grained components that can be tailored for communities of users. Our goal is to support Web accessible resources of pedagogical material that can be used and reused in a variety of contexts from teacher remediation and lesson preparation to the use of lessons that reflects the diversity of individual learning styles. The Web based lessons are generated on the fly from a database of fine grained components that include explanations, interactive activities, worksheets, images, videos, audio, FAQ's, online questions and challenges. This architecture had been used in the development of learning support systems for two different communities; users of middle school mathematics in one case and users of health informatics learning modules in the other case.

1. Introduction

In this paper we describe an architecture to support Web based lessons for online interactive learning. The architecture is based on a database of reusable and meta-tagged learning objects that can be dynamically combined to create personalized lessons, independent of the subject matter. A prototype has been developed based on this architecture for users (professionals, students and educators) of health informatics learning modules.

Typically, Web learning platforms provide either predefined packaged lessons or independent tools, such as applets or text, on a range of topics. Our goal is to move past this standard to support dynamic and flexible generation of lessons. This means that Web accessible resources of pedagogical material can be designed for use and reuse in a variety of contexts, from remediation to lesson preparation and use of the lesson. It also enables lessons to be tailored to individuals.

Individuals have characteristics that affect their learning success, such as learning styles, visual and textual cognitive preferences, and the ability to abstract (7, 11). Individual learning objects reflect this diversity. Therefore, each learning object is tagged on multiple dimensions; subject content, competencies addressed by this object, and the learning style for which this object is best suited. Similarly, the user's profile contains information relating to their learning style, competencies completed, modules completed, etc.

The architecture described in this paper supports the reuse of such tagged learning objects as well as the dynamic composition of objects into personalized lessons. Section 2 of this paper provides some background on learning modules and learning styles. Section 3 describes the core information architecture developed for the community of mathematics users. Section 4 describes this architecture embedded in a more complex software/hardware environment for the health informatics community and Section 5 provides a brief summary.

2. Background

Online modules must address a range of learning scenarios including distance learners, student reinforcement of material and parent review. Teachers may also use these resources to refresh their understanding of concepts before presenting a lesson or to modify a lesson. When a teacher is unsure of the material he or she is presenting, students pick up on these insecurities often creating a "non-productive learning atmosphere" (24).

Three major problems that persist in the static presentation of educational material are consistency, teacher effort and lesson generalization. When dealing with a significant number of content developers, inconsistencies may appear within the learning content

(16). For instance, two educators developing similar lessons may use different terminology that could result in inconsistent lesson content. Dividing content into reusable components allows educators to re-use existing material thus decreasing the introduction of inconsistent material. In addition to reuse, interoperability among tools, lesson content, databases and descriptions is also important (20). Secondly, requiring educators to create entirely new content has been found to be one of the major failings of earlier computer aided instruction systems (3). Teachers have stated that they have little time to learn new technologies and would rather adapt existing content instead of creating new content (21). By dividing text, image, audio, and other content into individual components, material can easily be reused and tailored for a specific lesson or classroom. Lastly, static course presentation is aimed at a general student audience, which may mean that a percentage of content is only relevant to a specific student group. Students for which a significant amount of material is not relevant, either because they do not have the necessary background knowledge or because they have already covered a concept, may experience disorientation (8), cognitive overload (18), and/or a less productive learning environment (19).

Learning styles can have a significant impact on a student's success in the classroom (10). Although students must be exposed to a variety of learning style modes to become successful students (11), they must also be given the opportunity to work in their preferred learning style mode. In recognition of this, online lessons should be able to offer a variety of learning objects that support these different learning styles. Active learners, for example, may prefer working with applets, where they can experiment for themselves, whereas reflective learners may prefer reading from a textual description. Partitioning content into individual learning objects allows lesson material to be dynamically personalized to meet these preferences.

Research in the area of adaptable and dynamic educational systems has grown in the past few years, taking full advantage of the concept of reusable learning objects. The iTeach system (22) adapts course content for individual students, presenting material via a dynamically created table of contents. Triantafillou, Pompuortsis and Georgiadou (23) have recently developed an online learning system that adapts for different cognitive styles. Conlan, et al., (6) have developed a meta-driven online learning system that adapts for student attributes such as prior knowledge, learner goals and learning styles. Adaptive navigation systems (4, 18) employ several different techniques that either hide or display links, depending on students' mastery of concepts, goals and prior knowledge. This approach could also be applied to individual components, instead of hyperlinks to create a personalized learning environment.

Our goal, then, is an information architecture that supports the pedagogy with technology to provide dynamic presentation and composition of lessons that are configurable and reusable across several dimensions including content type, learning style and content functionality.

3. Core Architecture

In this section we present the core architecture (15) that supports the dynamic generation of Web lessons for middle school mathematics using a variety of content types that appeal to a range of users. The content can address a variety of user needs, from remedial material to open challenges or enrichment material. A prototype was built based on this architecture. This architecture relies on a database of fine grained components that are tagged with a multidimensional feature set. The architecture is built on the three tier Web model, shown in Figure 1.

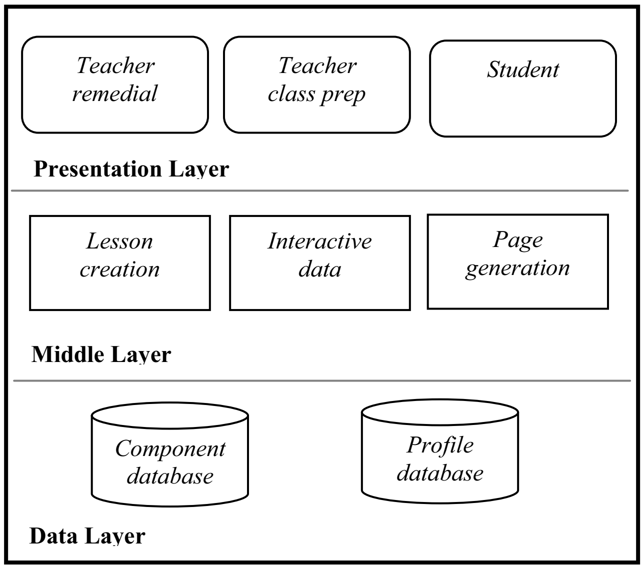


Figure 1. Core Architecture

3. 1. Data Layer

The data layer has two major parts: the content components and the user profile information. Content components are stored in the component database and consist of fine grained atomic units of learning material. These atomic units provide the maximum flexibility and reuse of these components in a variety of combinations. Each content unit is associated with a multidimensional feature set that determines its selection and use. Profile information is stored within a profile database and contains personal user characteristics including learning preferences, if any; usage data and histories and "handed in" material, where applicable.

3.2. Middle Layer

The middle layer is responsible for assembling the lesson material based on the content stored in the component and profile database. Lessons are created from the sequencing information stored in the component database. Interactive data such as student answers or new components are stored in the component and profile database respectively. All pages displaying lesson content are generated dynamically from the information stored in the database

3.3. Presentation Layer

The presentation layer supports the three intended user roles. Teachers can use lesson content to solidify their understanding of the material presenting the content to their class. Secondly, lessons can be modified by a teacher to reflect their students' abilities and learning styles. Lastly, the website can be used by students, either at home with their parents or in class, to review or complete coursework.

4. Embedded Architecture

A second prototype has been designed to implement the information architecture in a more complex environment to support a collaboratory of learning modules for health informatics available on the broadband CANARIE network across Canada. This prototype will support both distance and face-to-face learning. Rather than building stand-alone, domain-specific systems, we treat the core architecture as a modular structure, within which components can be replaced. For example, the learning objects can be stored within a relational database or within a more general learning object repository such as ALOHA (2), used by the Campus Alberta Repository of Educational Objects (5). In addition, the lessons produced in this system are packaged and compliant with the IMS Content Packaging Specification (13). This specification describes a standard method of organizing learning objects within an individual lesson.

Six independent health informatics lesson modules are in the final stages of completion by our module developers. The module developers have backgrounds that range from education to computer science to health informatics and medicine. The content developed for the prototype system follows the principles of constructivist learning (17) and was developed specifically for an e-learning environment. Some examples of the types of media used include movies created by the module developers, PowerPoint presentations synched with audio and video and standard html text and images. This content will be exported to a shared repository of educational learning objects.

Critical to the success of this architecture is the adoption of feature sets and vocabularies. The IMS Global Learning Consortium (12) has defined a set of specifications for e-learning systems. The IMS Metadata Specification (14) is used to describe learning objects and covers the information typically required of a learning object. The IMS tag set allows content developers to describe a number of educational features for a learning object which is useful when dynamically personalizing lesson content for individual student preferences as well as for educators searching for content that addresses specific educational needs.

The Advanced Distributed Learning (ADL) Initiative's "Sharable Content Object Reference Model" (SCORM) makes use of the specifications produced by IMS to define a "Web-based Content Aggregation Model and Run-Time Environment for learning objects." (1) Essentially, the SCORM standard defines how educational content is to be packaged so that it will be compatible with many learning management systems, how the content is to be displayed to the user and how the user's progress may be tracked.

The use of individual learning objects tagged according to the IMS Metadata and Content Packaging Specifications allows us to address the three problems presented in Section 2, above. The first problem is that of *Consistency*. The use of controlled vocabularies suggested by the IMS Metadata Specification insures a consistent description of all tagged learning objects. This allows teachers to search, reuse or adapt existing content instead of developing new and sometimes inconsistent lesson content. For the Health Informatics Collaboratory, Medical Subject Headings (MeSH) may be assigned to the objects as tags indicating the health informatics competencies the object is meant to address.

The second problem is that of *Teacher Effort*. With a database of meta-tagged learning objects, teachers can share common material that can be easily found using a search facility. Searches can be conducted based on a variety of IMS metadata tags. The use of a metadata standard such as the IMS specification also allows different repositories to pool their resources. This reduces the need for educators to create all material from scratch. Additionally, complete lesson packages can be reused in a Learning Management System (LMS) that supports the IMS Content Packaging Specification. As such, the lessons may be used directly in any IMS-compliant e-learning system such as WebCT (IMS Content Migration Utility), Blackboard or OpenText's Livelink.

The third problem is that of *Lesson Generalization*. The use of meta-tagged learning objects enables the personalization of individual lessons based on the metadata and how well they match a student's user profile and background.

4.1 Learning Object Hierarchy

A hierarchy of learning objects has been developed to allow teachers and module developers to reuse existing lesson content and is shown in Figure 2. All learning objects are tagged according to the IMS Metadata Specification. A learning module is an online lesson composed of an ordered list of pages. The hierarchy is composed of three abstract layers: Page, Component and Atom.

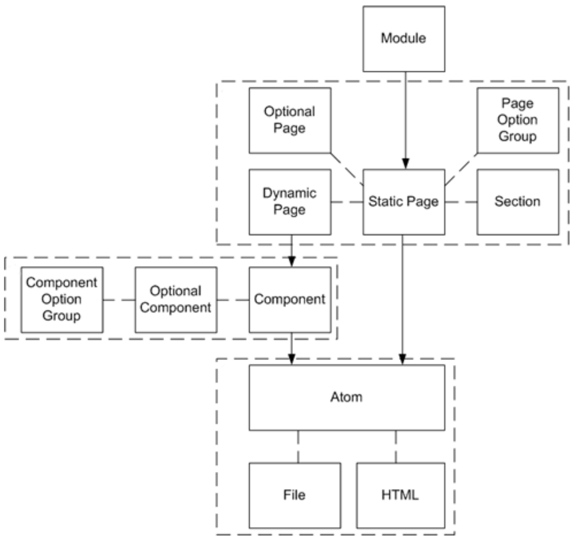


Figure 2. The learning object hierarchy

4.1.1. Page. A page represents a complete learning concept. Five types of pages are available, each having slightly different properties. A static page consists of a single file atom that is displayed as is. An example of static page is a page containing only single movie file. A dynamic page consists of an ordered list of components that are assembled to form an HTML page. An optional page is a single page, made up of an ordered list of components that will show or hide itself based on a set of parameters defined by a module developer. For instance, a module developer could define that a page containing remedial content be shown for students who have not completed a specific course while it is hidden for all other students.

A set of pages can be contained by a page option group. From this group, the most appropriate page is chosen for a student, using the student's profile and the learning objects' metadata. A section consists of an introduction page and an ordered list of pages. As well, sections can be composed of different types of pages. A section can also contain several subsections.

4.1.2. Component. Components serve as a reusable container for atoms that may appear together often, for instance, an image (file atom) and a caption (HTML snippet). Three types of components have been defined within the hierarchy: components, optional components and component option groups. A basic component is a container consisting of an ordered list of atoms that are put together on a page to form a coherent learning object. Optional components are displayed or hidden based on parameters defined by the module developer. Component option groups, similar to page option groups, contain a set of components from which the most appropriate component is selected for the student.

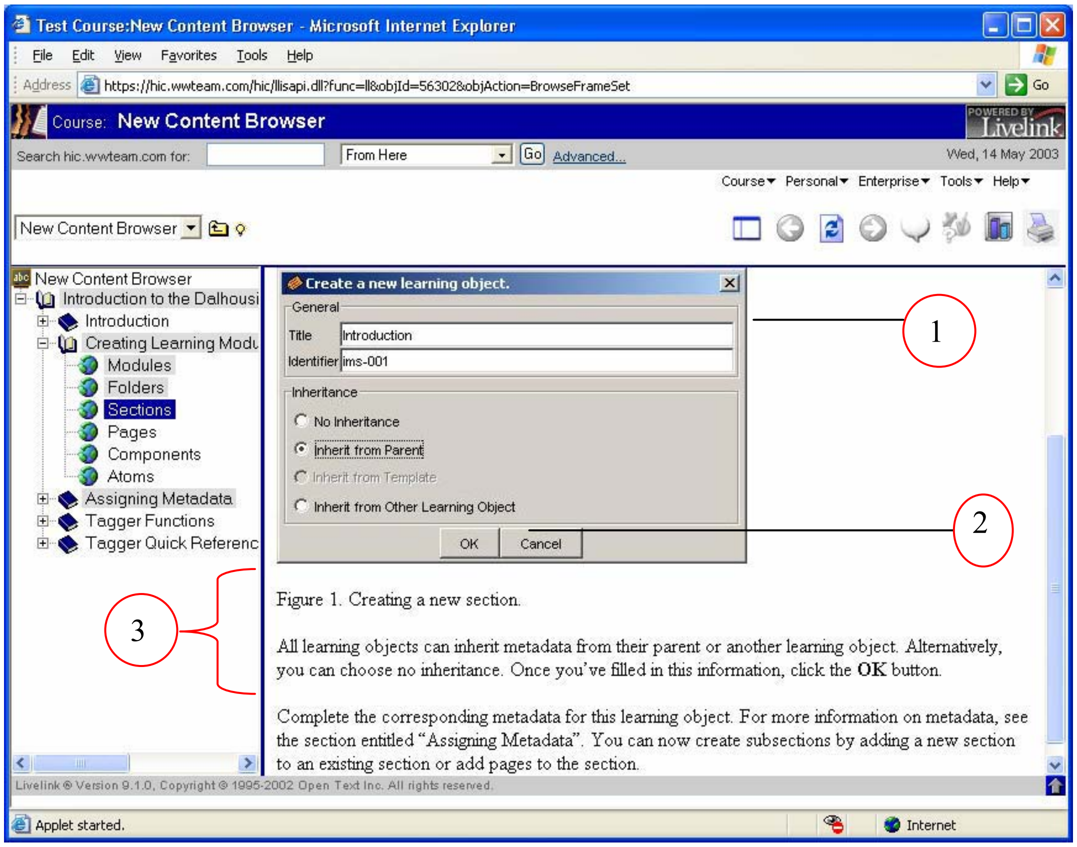
4.1.3. Atom. An atom is the smallest unit of learning content. Two types of atoms have been defined: file atoms and HTML atoms. A file atom can be composed of a single file such as a single PowerPoint slide, an image or a movie clip. An HTML atom consists of a snippet of HTML code, for instance a caption, paragraph, table or list. Atoms can be grouped with other atoms to form components.

A dynamically generated page from an example module is shown in Figure 3 through the Livelink Content Browser. This page was generated dynamically to teach users how to create learning objects. The image labeled "1" is a file atom and the caption describing the image, labeled "2", is an HTML atom. These two atomic learning objects are combined together to form a single reusable component. The two paragraphs labeled "3" are a single textual atom and may be combined with other atoms to form a component

4.2. Learning Object Management System

This architecture supports a Canada-wide learning collaboratory of modules for health informatics education. Figure 4 shows the architecture described in Section 3, embedded in a more complex system called the Learning Object Management System (LOMS). The component and profile databases have been replaced by the Learning Object Store, a repository for lesson components, metadata and sequencing information. The middle layer controls access to the learning objects, selects objects appropriate to the competencies of a user and creates customized educational content to be displayed by the e-Learning Management System.

A tagger has been developed that allows content developers to easily tag learning learning objects. For this system, we use the OpenText Corporation's Livelink Content Browser to display lessons packaged using the IMS Content Packaging Specification. Livelink provides instructors with the functionality to deploy educational content and provides features such as activity tracking, grade management and communication channels.


Test Course:New Content Browser - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address https://hic.wwteam.com/hic/llisapi.dll?func=ll&objId=56302&objAction=BrowseFrameSet
Course: New Content Browser
Search hic.wwteam.com for: From Here Go Advanced...
Wed, 14 May 2003
Course Personal Enterprise Tools Help
New Content Browser
Introduction to the Dalhousi
Introduction
Creating Learning Modu
Modules
Folders
Sections
Pages
Components
Atoms
Assigning Metadata
Tagger Functions
Tagger Quick Referenc
Create a new learning object.
General
Inheritance
OK Cancel
1
2
3
Livelink ® Version 9.1.0, Copyright © 1995-2002 Open Text Inc. All rights reserved.
Applet started.
Internet

Title: Introduction
Identifier: ims-001

No Inheritance
[✓ Inherit from Parent]
Inherit from Template
Inherit from Other Learning Object

Figure 1. Creating a new section.

All learning objects can inherit metadata from their parent or another learning object. Alternatively, you can choose no inheritance. Once you've filled in this information, click the **OK** button.

Complete the corresponding metadata for this learning object. For more information on metadata, see the section entitled "Assigning Metadata". You can now create subsections by adding a new section to an existing section or add pages to the section.

Figure 3. A dynamically generated lesson displayed in the Livelink Content Browser.

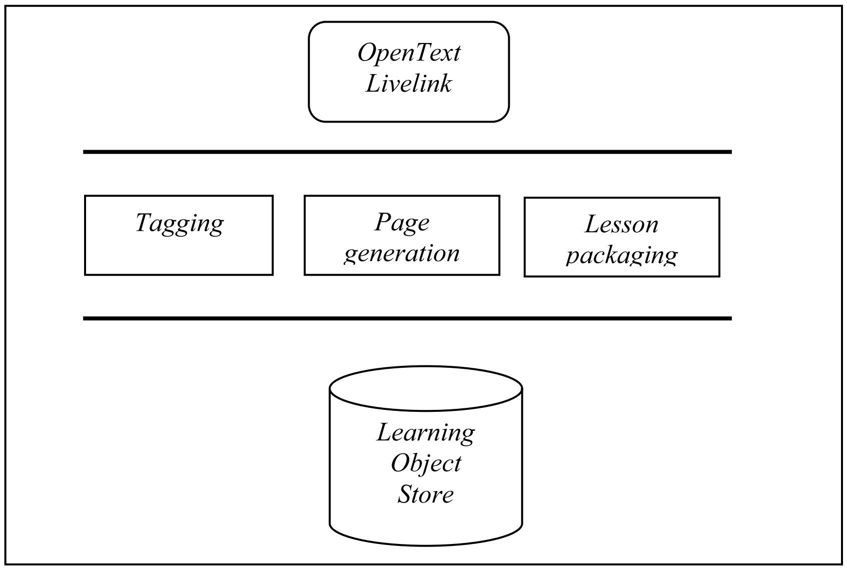


Figure 4. The embedded architecture

Three main types of user groups are supported within this architecture: content developers, course instructors and students. The content developers are responsible for creating, tagging and packaging the lesson content. Course instructors, who may or may not be content developers themselves, are responsible for importing the appropriate modules into Livelink to form the desired lessons. The health informatics students will access the lesson material through the Livelink Content Browser and interact with their instructor and other students using the many communication channels offered by Livelink. In addition, the students themselves may also import desired modules into Livelink to form desired lessons for remedial work or for further exploration.

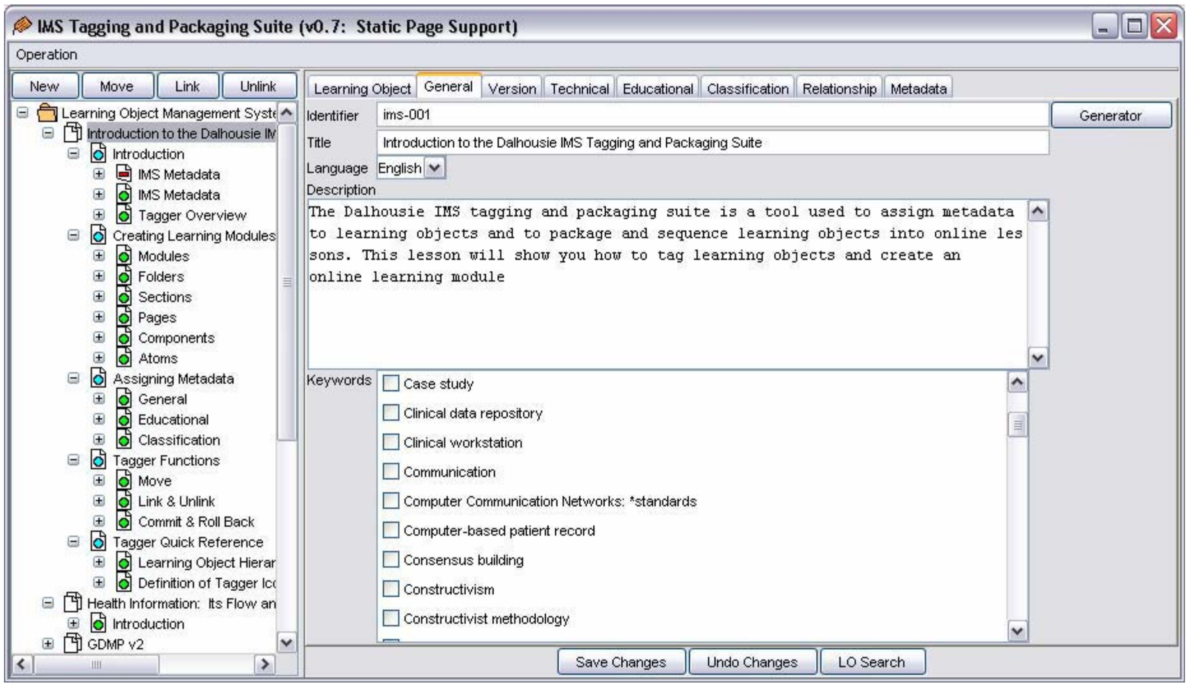
4.2.1. Learning Object Store. The Learning Object Store is a relational database that stores the learning objects, their metadata and student profile information. All file data such as binary data, text and HTML snippets and files for the atoms are stored within the database. In addition to IMS metadata, a learning object is also associated with a set of health informatics competencies and MeSH subject headings. The sequence of learning objects defined by the content developer in the tagger is stored in the database. This defines the order of the pages presented and the order of the components and atoms on the page.

The system is designed to handle student profiles which would also be stored in the learning objects database. In addition to personal information such as name, location and time zone, the student profile will contain health informatics competencies that have yet to be met in order to satisfy program requirements, the competencies that have already been met and a listing of modules completed. An attempt has been made to assign as many of the same characteristics to a student as a learning object in order to facilitate lesson personalization.

4.2.2. Tagging. A tagging package supporting the IMS Metadata Specification has been developed and is shown in Figure 5. Content developers are the only user group with access to the tagger. This software allows content developers, many of whom do not have the expertise to author XML documents, to easily create metadata describing individual and groups of learning objects, sequence objects within a lesson and define any dynamic aggregation. Content developers can develop their content externally and use the tagger to assign metadata descriptions as the content is added to the repository.

The design for the tagger centered on the need to quickly and easily create metadata for learning objects. Tags for which data can be generated automatically, for instance file size or file type are automatically filled in. Metadata are input using a mix of textfields, drop-down menus and tree structures. An attempt has been made to reduce the amount of "free text" entry to ensure that the metadata conforms to the suggested vocabularies. We have taken advantage of the IMS best practice vocabulary and guidelines to populate the input structures. Reducing inconsistencies within the metadata improves interoperability among different repositories.

Within the tagger, a content developer can create a new module and all of the objects defined within the hierarchy. Learning objects can be moved, linked to other learning objects and unlinked from a parent. Learning objects can inherit metadata from their parents which significantly reduces the amount of time required to properly tag all the learning objects in a lesson. Using the inheritance feature, templates can be created to ease creation of tagged objects. Once metadata tagging is complete, all binary files, HTML snippets and files are uploaded to the Learning Object Store through the tagger.


IMS Tagging and Packaging Suite (v0.7: Static Page Support)

Identifier: ims-001

Title: Introduction to the Dalhousie IMS Tagging and Packaging Suite

Language: English

Description:

```
The Dalhousie IMS tagging and packaging suite is a tool used to assign metadata
to learning objects and to package and sequence learning objects into online les
sons. This lesson will show you how to tag learning objects and create an
online learning module
```

Figure 5. IMS Tagging and Packaging Suite. Notice that the module currently being tagged is the same as that shown in Figure 3 in Livelink's Content Browser.

4.2.3. Lesson Packaging & Page Generation. Each lesson within the LOMS is packaged according to the IMS Content Packaging Specification. However, packaging depends on the page types defined within a module. A module without any optional or grouped learning objects (e.g., optional pages or component option groups) is packaged once for all students and is imported into Livelink by the course administrator.

Some lessons within the LOMS have been designed to include and exclude remedial content. Based on responses to a self-assessment, the LOMS suggests whether specific parts of the lesson should be included or excluded and gives a justification.

To create a personalised lesson, the content creator creates four sets of definitions: the competencies, the optional learning objects that are to be activated and deactivated, the questionnaire and the rule set for activating and deactivating learning objects.

The competencies are scalar variables storing the user's level of competence in a specific subject area. The content creator is free to use and mix any level of granularity for the competencies that they need. For example, the content creator could define a competency for "general computer skills" that can be related to something more specific, such as "keyboarding skills."

Associated with the set of learning objects to be activated or deactivated are justifications for why the object may be enabled or disabled based on the user's competency level. This allows the content creator to give specific, meaningful justifications for why the user may or may not be interested in the particular piece of content.

A competency self-assessment is defined by the content creator. The self-assessment consists of a series of questions that may be multiple choice or true or false or may be a self-assessment using a Likert scale. Each answer is associated with one or more competencies and influences their associated scalar values.

Once the user completes their self-assessment, all of the content recommendations are presented to the user with a justification. At this point, they can override any of the decisions made by the program. When the user is ready, the changes are made to the lesson and it is packaged using the method described above.

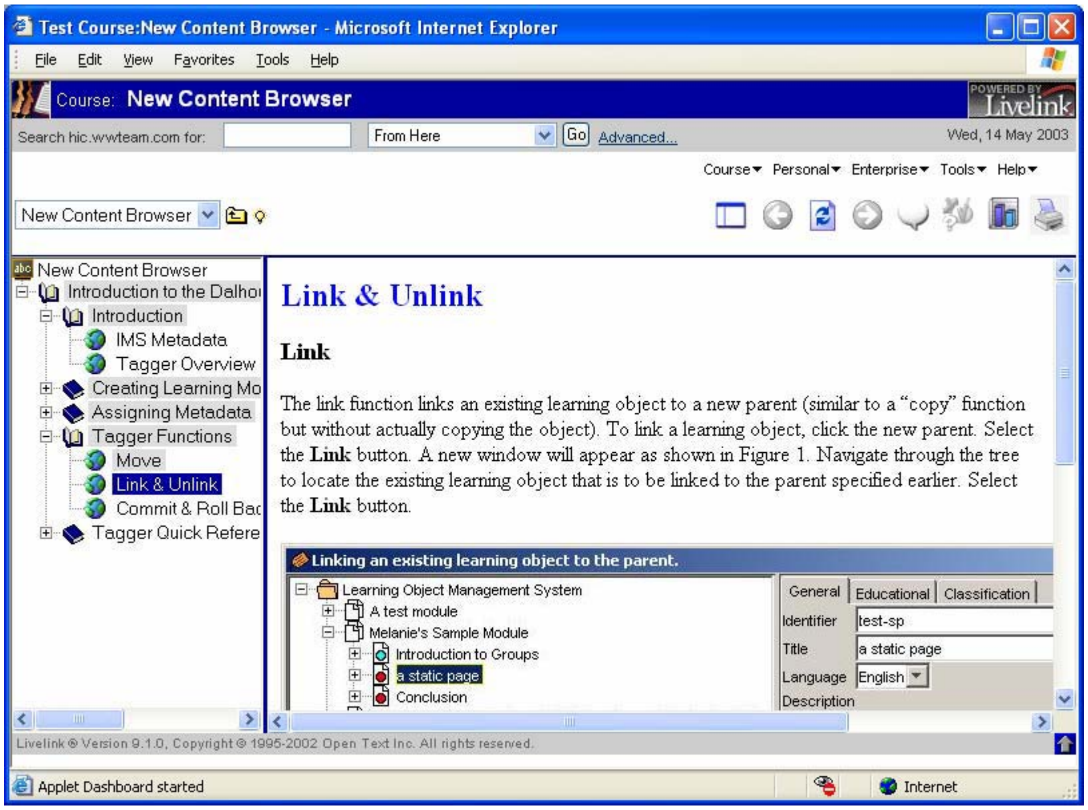
Since it is infeasible to create the hundreds of content packages that would be needed for each student to receive personalized modules, the content package does not contain the actual lesson content. It contains a series of links to the LOMS, containing pointers to the pages and atoms in the Learning Object Store. This significantly reduces the amount of storage necessary. The personalized lessons are generated dynamically by the LOMS at runtime.

4.2.4. OpenText Livelink. The Livelink e-Learning Management System is responsible for all interactions with both teachers and students, such as lesson presentation and grade tracking. The generated lessons are presented to the student through Livelink's Content Browser as shown in Figure 6. The left hand side bar displays a table of contents that has been generated by the IMS Content Package for this particular module. The sections and pages are displayed in the table of contents and are marked as visited once the student has opened the page. In addition to lesson presentation, Livelink also includes several forms of computer-mediated communication (CMC), such as discussion boards, whiteboards and chat, which enable collaborative learning among students.

5. Summary

The architecture that has been developed for this project supports the reuse of learning objects and, as such, addresses the issues of consistency, teacher effort and lesson generalization as discussed above. We have developed a prototype system based on this architecture for health informatics education. The prototype allows module developers to tag learning objects according to the IMS meta-data specification and aggregate them together to form a learning module. Lesson content is delivered through OpenText's Livelink e-Learning Management System. Personalized lessons are delivered to students based on their knowledge of required health informatics competencies. Optional learning objects are either included or excluded from the module and a justification for the decision is provided to the student.

The Health Informatics Collaboratory project is a CANARIE Inc. funded research project. In this project, six learning modules in health informatics have now been completed. All of the modules were created using atomic-level components that were tagged as described in this paper.


Test Course:New Content Browser - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Course: New Content Browser
POWERED BY Livelink
Search hic.wwteam.com for:
From Here
Go
Advanced...
Wed, 14 May 2003
Course Personal Enterprise Tools Help
New Content Browser
New Content Browser
Introduction to the Dalhou
Introduction
IMS Metadata
Tagger Overview
Creating Learning Mo
Assigning Metadata
Tagger Functions
Move
Link & Unlink
Commit & Roll Bac
Tagger Quick Refere
Livelink ® Version 9.1.0, Copyright © 1995-2002 Open Text Inc. All rights reserved.
Applet Dashboard started
Internet

Link & Unlink

Link

The link function links an existing learning object to a new parent (similar to a "copy" function but without actually copying the object). To link a learning object, click the new parent. Select the **Link** button. A new window will appear as shown in Figure 1. Navigate through the tree to locate the existing learning object that is to be linked to the parent specified earlier. Select the **Link** button.

Linking an existing learning object to the parent.

Learning Object Management System
A test module
Melanie's Sample Module
Introduction to Groups
a static page
Conclusion

General | Educational | Classification

Identifier: test-sp
Title: a static page
Language: English
Description

Figure 6. Livelink Content Browser displaying a learning module.

An evaluation of the completed modules is now underway. Training sessions have been held for all module developers at each site. Most module developers found the tagger easy to learn and simple to use. The most popular feature of the tagger was the template functionality because it significantly reduced the amount of tagging required. The biggest hurdle we encountered was convincing module developers to think of their module as a collection of learning objects and not a single learning object. This was a substantial paradigm shift for many developers and some were able to make that shift much easier than others.

6. Acknowledgements

We wish to acknowledge the support of CANARIE, Inc., Ottawa, and of Open Text Corporation, Waterloo, Ontario, Canada.

7. References

[1] Advanced Distributed Learning (ADL). (2002). SCORM Overview. Retrieved January 22, 2003, from http://www.adlnet.org/index.cfm?fuseaction=scormabt

[2] ALOHA. (2002). ALOHA Streams of Knowledge. Retrieved January 22, 2003, from http://aloha.netera.ca/

[3] Baloian, N., Pino, J. & Hoppe, H. (2000). A Teaching/Learning Approach to CSCL. In Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, January 4-7. New York: IEEE. [CD-ROM, 10 pages].

[4] Calvi, L. & De Bra, P. (1997). Improving the Usability of Hypertext Courseware through Adaptive Linking. In Proceedings of the ACM Conference on Hypertext, Southampton, UK, April 6-11. 224-225.

[5]Campus Alberta Repository of Educational Objects (CAREO). (2002). Repository in a Box. Retrieved January 22, 2003, from http://careo.netera.ca/

[6] Conlan, O., Wade, V., Bruen, C. & Gargan, M. (2002). Multi-model, Metadata Drive Approach to Adaptive Hypermedia Services for Personalized eLearning. In P. De Bra, P. Brusilovsky and R. Conejo (Eds.): AH 2002, Málaga, Spain, May 29-31. (Lecture Notes in Computer Science 2347) Berlin: Springer. 100-111.

[7] Dunn, R. & Dunn, K. (1978). Teaching Students Through Their Individual Learning Styles: A Practical Approach. Virginia: Prentice-Hall.

[8] Eklund, J. & Sinclair, K. (2002). An Empirical Appraisal of the Effectiveness of Adapative Interfaces for Instructional Systems. Educational Technology & Society. 3(4): 165-177.

[9] Ford, N. (1995). Levels and Types of Mediation in Instructional Systems: an Individual Differences Approach. International Journal of Human-Computer Studies. 43(2):241-259.

[10] Felder, R. (1995). Learning and Teaching Styles In Foreign and Second Language Education. Foreign Language Annals. 28(1):22-31.

[11] Felder, R. (1996). Matters of Style. ASEE Prism. 6(4):18-23.

[12] IMS Global Learning Consortium. (2003). Retrieved January 30, 2003, from http://www.imsglobal.org/

[13] IMS Global Learning Consortium. (2001a). IMS Content Packaging Specification. Retrieved January 22, 2003, from http://www.imsglobal.org/content/packaging/

[14] IMS Global Learning Consortium. (2001b). IMS Learning Resource Meta-data Specification. Retrieved January 22, 2003, from http://www.imsglobal.org/metadata/

[15] Kellar, M., MacKay, B., Zhang, R., Watters, C., Kaufman, D. & Borwein, J. (2003). Architecture to Support Dynamic Composition of Math Lesson Plans. In Proceedings of the 2003 ACM Symposium on Applied Computing, Melbourne, Florida, March 9-12. New York: ACM. 569-574.

[16] Kostur, P. (2002). Connecting Learners with Content: A Unified Content Strategy for Learning Materials. In Proceedings of the 20th Annual International Conference on Computer Documentation, Toronto, Canada, October 22-23. New York: ACM. 100-103.

[17] Jonassen, D. (1999). Constructivist learning environments on the web: engaging students in meaningful learning. Paper presented at the Educational Technology Conference and Exhibition, Singapore. Retrieved September 24, 2003 from http://www.moe.edu.sg/iteducation/edtech/papers/d1.pdf.

[18] Pilar da Silva, D., Van Durm, R., Duval, E. & Olivie, H. (1998). Adaptive Navigational Facilities in Educational Hypermedia. In Proceedings of the ACM Conference on Hypertext, Pittsburgh, Pennsylvania, June 20-24. New York: ACM. 291-292.

[19] Pawlowski, J.M. (2002). Reusable Models of Pedagogical Concepts - a Framework for Pedagogical and Content Design. In Proceedings of ED-MEDIA 2002, World Conference on Educational Multimedia, Hypermedia & Telecommunications, Denver, Colorado, June 23-28. Retrieved January 27, 2003, from http://beta1.wi-inf.uni-essen.de/research/publications/pawlowskiedm2002.pdf

[20] Roschelle, J., DiGiano, C., Chung, M., Repenning, A., Tager, S. & Treinen, M. (2000). Reusability and Interoperability of Tools for Mathematics Learning: Lessons from the ESCOT Project. In *Proceedings of Intelligent Systems & Applications* at University of Wollongong, Australia. 664 – 669.

[21] Rosson, M.B. & Seals, C. (2001). Teachers as Simulation Programmers: Minimalist Learning and Reuse. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, Washington March 31–April 5. New York: ACM. 237-244.

[22] Seeberg, C., Steinacker, A., Reichenberger, K., Fischer, S. & Steinmetz, R. (1999). Individual Tables of Contents in Web-based Learning Systems. In Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia, Darmstadt, Germany, February 21-25. New York: ACM. 167-168.

[23] Triantafillou, E., Pomportsis, A. & Georgiadou, E. (2002). AES-CS: Adaptive Educational Systems based on Cognitive Styles. In P. Brusilovsky, N. Henze and E. Millán (eds): Proceedings of the Workshop on Adaptive Systems for Web-based Education, held in conjunction with AH'2002, Málaga, Spain, May 29-31. 1-12.

[24] Wu, H. (1997). On the Training of Mathematics Teachers. Retrieved August 14, 2002, from http://math.berkeley.edu/~wu/teacher-education.pdf