Engineering a Future for Web-based Learning Objects

Permanand Mohan[1] and Christopher Brooks[2]

[1] Department of Mathematics and Computer Science, The University of the West Indies,
St. Augustine, Trinidad and Tobago
{pmohan@tstt.net.tt}
[2] ARIES Lab, Department of Computer Science, University of Saskatchewan,
Saskatoon, Canada
{cab938@mail.usask.ca}

Abstract. This paper takes a critical look at current development efforts with learning objects for Web-based e-learning. It points out the limitations of these efforts and argues that they are still a long way off from realizing the potential of learning objects on the Web. The paper then proposes the notion of object-oriented learning objects to address several of the problems identified.

1 Introduction

One aspect of e-learning currently receiving considerable attention is the *learning object* — an entity of learning capable of being reused in different instructional situations. Learning objects now seem poised to fill the ever-increasing need to quickly develop cost-effective training materials for e-learning. Recently, a number of standardization efforts have resulted in several specifications for e-learning and learning objects. These efforts are being spearheaded by groups such as the Learning Technology Standards Committee of the Institute of Electrical and Electronics Engineers (IEEE LTSC) [ltsc.ieee.org] and the IMS Global Learning Consortium [www.imsglobal.org]. These groups have developed specifications to aid in the discovery, management, and exchange of learning objects.

The IEEE 1484.12.1-2002 Standard for Learning Object Metadata (LOM) is the first accredited standard for learning technology. It is essentially a cataloging scheme for learning objects and uses nine categories of metadata elements to describe a learning object. Two important specifications from the IMS are Content Packaging and Simple Sequencing. The former allows learning objects for an individual course or collection of courses to be packaged into interoperable packages, while the latter allows the sequencing of learning objects within a content package.

Different types of repositories are also being developed to allow users to discover, obtain rights to, and use learning objects on the Web. These include global repositories that are based on the client/server approach, repositories employing a brokerage model, and local repositories that provide peer-to-peer access to local repositories of learning objects.

2 Limitations of Current Efforts with Learning Objects

LOM metadata allows searching for learning objects based on keywords and other basic pedagogic metadata such as *LearningResourceType*, *TypicalAgeRange*, or *Language*. To determine the suitability of a learning object, an instructional designer must carefully examine each learning object. This can be very time consuming, given that many learning objects may satisfy the query and each one must be examined individually. The Digital Repositories Interoperability (DRI) Specification from the IMS makes it easier for computers to query on-line repositories for learning objects. However, the problem still remains, since the metadata is the same and computers cannot perform the level of filtering required.

Making instructionally principled decisions with learning objects requires deep pedagogic information in the metadata. Current learning object metadata say nothing about how to combine learning objects with others, and indeed, whether such a combination is useful in the first place. Metadata also say nothing about the types of learners for which a learning object is best suited and the kinds of teaching and learning strategies it employs. Worse yet, metadata do not even identify the learning outcome(s) that will be achieved if the content in a learning object was successfully learned [1]. Putting together content packages based on reusable learning objects is a very difficult problem. However, this problem must be solved otherwise it will severely limit the usefulness of the numerous repositories of learning objects that are being developed.

In addition to the problem of locating the 'right' learning objects, the full set of 86 elements in the IMS Metadata Specification is not suited to direct implementation since it entails a huge classification effort [2]. Also, widely varying interpretations of the utility, scope and purpose of individual elements threaten to cause considerable interoperability problems, which the specification was designed to solve in the first place. Thus, even if the metadata specifications were very useful in automatically generating content packages (and they are not), creating the metadata itself entails a huge effort.

We therefore contend that the future of learning objects is at stake. Current development efforts do not address the really important issues associated with using learning objects for e-learning. There is much excitement about metadata, the Semantic Web, RDF, ontologies, and other technology that promise a bright future for locating information on the Web. These technologies are poised to profoundly influence the way in information is used in the future Web. However, they do not address many of the problems related to successfully using learning objects in an instructionally meaningful manner.

3 Nature of Learning Objects

It is often suggested that learning objects are really like the objects of object-oriented software [e.g., 3, 4]. We now examine this claim. Consider audio-visual media files and files of other formats (e.g., GIF, JPEG, PowerPoint Presentations, etc.) which are simply collections of bits. These files are not software, much less software objects. So, it is clear that not every learning object is software. However, files such as HTML files containing markup and code written in scripting languages such as JavaScript and Java applets may be considered software. However, HTML files do not support object-oriented concepts, so are not object-oriented. Java applets are certainly software. From the perspective of its developers, a Java applet may be considered object-oriented. From the perspective of an instructional designer or other programmer wishing to use the applet (without access to the source code), the applet is just like another piece of software, and indeed, its object-oriented features are irrelevant.

Thus, despite the claims being made, a cursory examination reveals that most learning objects today are not truly object-oriented from the computer science perspective. However, from an instructional designer's point of view, there is no real benefit in viewing learning objects as object-oriented and it is best to look at learning objects in terms of their content, context of use, and relationship with other learning objects. To require one to understand object-oriented theory to put together meaningful content packages is perhaps akin to asking a good carpenter to understand the internal structures of the materials that are used to fabricate furniture.

From a computer science perspective, there is much to gain by treating learning objects as object-oriented software artifacts. Object-oriented technology can be used to take learning objects out of their current static form and imbue them with behaviours that allow them to contribute more meaningfully to an instructional situation. Object-oriented learning objects are capable of overcoming several of the limitations mentioned in the previous section. Moreover, instructional designers and other users do not have to be aware of this technology, much like users of Microsoft Office are not aware of the agent-like technology underlying the Office Assistant.

4 The Object-Oriented Learning Object

In this section, we propose the idea of an object-oriented learning object. There is one naturally occurring class of learning objects, *LearningObject*, which is the superclass of all learning objects. Every learning object is an instance of this class. *LearningObject* has properties such as a collection of references to metadata instances (e.g., Dublin Core, CanCore, or LOM). *LearningObject* also contains a set of different *Version* objects that refer to different versions of the learning resources that make up the learning object. An important property is a *Context* object that contains various types of contextual information such as the types of learners for which the learning object is appropriate and the teaching and learning strategies employed by the learning object. Another property is a *Combination* object, which contains information about the characteristics of the learning object that allows it to be combined with other learning objects. *LearningObject* also contains one or more concept maps, describing how the learning object is positioned in the conceptual structure of a domain, and the instructional objectives that will be achieved if the content is successfully learned.

LearningObject provides several useful methods. Query methods accept a *Context* object as well as one or more *Metadata* objects and determine if the learning object is appropriate. Version control [5] is achieved by means of various methods operating on the list of *Version* objects. Since a learning object can contain several learning resources or even aggregate other learning objects, *LearningObject* has methods to enable the easy insertion, deletion, and rearrangement of learning resources within a learning object. This gives content producers (and others) the flexibility to easily make changes to a learning object from time to time. Methods can also be written to render the learning object in different formats, e.g., HTML, text, PDF, XML, etc. In essence, *LearningObject* has meaningful data for reasoning about instruction, and methods to support the reasoning process and the increased complexity of the object-oriented learning object.

Since learning objects are independent chunks of content, it is difficult for them to have *using* relationships with other learning objects. *Using* relationships introduce undesirable coupling between learning objects, preventing them from standing on their own, and decreasing the possibility of reusability. However, *aggregation* relationships allow hierarchies of learning objects to be created out of simpler ones. *Inheritance* is also very useful since it is possible to inherit metadata and other attributes of a learning object when producing similar learning objects. This reduces considerably the data entry required for entering metadata and other information.

We believe that the object-oriented approach can go a long way towards achieving the vision currently being promoted for learning objects. Learning objects with object-oriented features provide a solid foundation for the effective reuse of learning resources on the Web.

References

1. Mohan, P., and Greer, J. (2003). E-Learning Specifications in the Context of Instructional Planning. Accepted for the International Conference on Artificial Intelligence and Education, Sydney, Australia, July 2003.
2. Richards, G. R., McGreal, R., and Freisen, N. (2002). Learning Object Repository Technologies for TeleLearning: The evolution of POOL and CanCore. Proceedings of the Informing Science + IT Education Conference, Cork, Ireland, June 19-21, 2002, p. 1333-1341.
3. Downes, S. (2001). Learning objects: resources for distance education worldwide. *International Review of Research in Open and Distance Learning*, 2 (1).
4. Robson, R. (1999). Object-oriented Instructional Design and Applications to the Web. World Conference on Educational Multimedia, Hypermedia and Telecommunications, Seattle, Washington, USA, June 19-24, 698-702.
5. Brooks, C., Cooke, J., and Vassileva, J. (2003). Versioning of Learning Objects. Accepted for the International Conference on Advanced Learning Technologies, Athens, Greece, July 2003.