

Tree View Editing Learning Object Metadata

Zeynel Cebeci
Cukurova University in Adana,
Turkey

Yoldas Erdogan
Cukurova University in Adana,
Turkey

zcebeci@cukurova.edu.tr

yerdogan@cukurova.edu.tr

Abstract

This paper introduces and examines an authoring tool called as “TreeLom” for producing the metadata compatible to IEEE LOM draft standard. TreeLom, has been developed with MS .NET framework technology, is an application of XML binding of the LOM. Its tree view editing interface provides rapid data input in building learning object metadata.

Keywords: Learning objects, Learning object metadata, XML binding, LOM editor, Tree view editors

Introduction

For the e-learning community, the “learning object (LO)” has emerged as the most dominant concept among various approaches and methods in asynchronous Web based teaching and learning (WBT/WBL). The IEEE Learning Technology Standards Committee (IEEE LTSC) defines that a learning object is “any entity, digital or non-digital, which can be used, reused or referenced during technology supported learning” (IEEE LTSC, 2002). However Wiley (2003) excludes non-digital entities in his definition, and defines a learning object as “any digital resource that can be reused to support learning”. In fact, the term “learning objects” generally applies to the digital units of educational content designed and created for many purposes such as maximizing the number of learning situations, reducing development costs, and exchanging of learning materials between different platforms. LOs can be any electronic file format such as text, image, graphics, audio, video or an interactive program.

The popularity of LOs is due to their main features such as reusability and interoperability (or portability) in addition to many other secondary or derivative features that may be used to design them such as granularity, accessibility, manageability and durability. In order to utilize them for building courses, LOs should be available in a learning object repository (LOR) or a learning system. LORs are the systems that designed to store, search, retrieve, and deliver LOs to learners. The retrieved LOs are aggregated to construct whole learning content in the forms of lessons,

modules and courses. In Learning Management Systems (LMS) and/or Learning Content Management Systems (LCMS) that have recently been used in WBT/WBL, same learning object can be reused to build different e-learning content or online courses because LOs are independent and small sized components of knowledge (or chunks of learning information). Thus, LOs can be also called as “reusable learning objects (RLO)” and

Material published as part of this journal, either on-line or in print, is copyrighted by the publisher of the Informing Science Journal. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@ijklo.org to request redistribution permission.

“shareable objects” because of their reusability for different educational disciplines, classes and purposes.

In addition, LOs are easily accessible, searchable, and portable objects in-and-between many operating platforms and/or LMS/LCMSs. All these features of LOs are basically provided with their metadata. Since metadata is “data about the data”, learning object metadata (LOM) can be simply defined as “descriptive data about learning objects”. Anido, Rodríguez, Caeiro, & Santos (2003) stated, “As available educational resources grow and grow the need of metadata becomes apparent. The lack of information about the properties, location or availability of a resource could make it unusable”. Therefore developing a metadata standard for describing LOs became a key issue.

The IEEE Learning Technology Standards Committee (LTSC) has provided intensive support for the development and maintenance of the Learning Object Metadata standard since 1997. The works on LOM done by LTSC LOM Working Group (WG12) with the members representing more than 15 countries have been resulted in an IEEE-accredited standard as the first part of a multi-part standard in June 2002 (IEEE 1484.12.1-2002 (IEEE LTSC, 2002)). Although the final draft can be downloaded from the LTSC’s Web site (IEEE LTSC, 2004) the approved version of the standard is not yet available due to minor editing and formatting works. According to the draft standard, the relevant characteristics of a LO are described with its metadata can be grouped into nine categories such as general, life cycle, meta-metadata, educational, technical, educational, rights, relation, annotation, and classification. The standard specifies the syntax and semantics of learning object metadata, defined as the attributes required to fully and describe a learning object which includes element names, definitions, data types, taxonomies, vocabularies, and occurrences and size limitations of the fields.

IEEE 1484.12.1-2002 is the first part of a multi-part standard that aims to facilitate the search, evaluation, acquisition, and use of learning objects by learners, instructors, or automated software processes (IEEE LTSC, 2002). At the present, the other parts of the standard intend to integrate LOM for XML, RDF and ISO/IEC 11404. Among these, the standard for XML binding for LOM data model (IEEE P1484.12.3/D2) is currently in the stage of Draft Version 2, and can be retrieved from the LTSC’s Web site. IEEE LTSC (2004, Abstract Paragraph) explains that “The purpose of this standard is to allow the creation of LOM instances in XML. This allows for interoperability and the exchange of LOM instances between various systems. This standard uses the W3C XML Schema definition language to describe the syntax and semantics of XML encodings.”

LOM provides learners, teachers, course authors or content developers a way to easily and quickly find LOs by searching digital learning objects repositories. Hence the use of LOs will reduce development costs and prevent duplicative efforts or works to build a course. Generally, LOs are reviewed by subject experts based on technical and educational evaluation criteria, so high-quality materials will be produced for publishing via learning systems. A dozen general purpose and discipline specific LORs which use LOM are already in operation. MERLOT, CAREO, iLumina, and Learning Matrix are several resources for LOs. It is expected that the LORs will increasingly contribute to the intensive use of the LOs in institutional e-learning systems.

There is a growing demand to the implementations of LOM for different platforms and/or technologies. Whereas LOM is an essential key in LO concept and technology the works on LOM authoring systems and tools are still limited in number and diversity. Hodgins (2000, p.19) also pointed to a critical need for “raising awareness, education, dissemination, and the tools and technology” which to start implementing the LOs.” In a Learning Technology Roundtable Panel Discussion organized by ISO/IEC JTC1 SC36, Lindner (2001) reported that promotion of learning technology standards should take high priority and urgency, and the action should start with easy

access to information on standards activities by repositories, explanatory documents, and translations. He also suggested that encouraging community with publicly available tools (e.g. LOM generator) should be considered. Similarly, Synytsya (2003, para. 1) reported that “the IEEE LOM standard and a number of recommended specifications of educational metadata are serving to facilitate to the development of large-scale distributed learning frameworks and new models of learning resources design and delivery, but have a limited implementation experience behind them”.

The discussions about the future of LO technology clearly indicate that the e-learning community needs the tools to build learning objects and their metadata in order to make them interoperable between systems and/or institutions. Therefore, this study aimed to discuss an easy-to-use online tool as an application of XML binding for LOM. Because the majority of existing tools and systems have been developed with Java technology, in this study the LOM editor is based on MS .NET framework technology for native adoption and integration into any LMS/LCMS that runs under the Windows platforms.

Editing Tools and Systems for LOM

Learning object metadata are created and edited with the software called “metadata editors” or “metadata generators”. These tools can be a stand-alone or an integral part of a system, and they differ in their functionalities and efficiencies for editing metadata. Several commercial and open-source freeware tools have been developed in order to edit and manage LO metadata files since the first publication of the specifications for LOM came into existence. In this part of the study, existing LOM editors and system have been reviewed to understand the technology and approaches used in their development.

As an early developed editor, the IMS Editor Vimse (ImseVimse) is one of the well known LOM editors. The ImseVimse has been expressed as a full-fledged graphical IMS metadata editor written in Java, and tested with JDK1.2.2 and 1.3.0 (Nilsson & Palmér, 2002). It is a product from the Conzilla project, a first prototype of a concept browser developed in the Knowledge Management Research (KMR) Group at the Centre for User Oriented IT Design, The Royal Institute of Technology in Stockholm, Sweden. The source code of ImseVimse is distributed under the GPL, and packages for Windows and X11 platforms can be downloaded at Sourceforge.Net site (<http://www.sourceforge.net/projects/imsevimse>). The current version of ImseVimse (v 0.9) supports IMS metadata specification version 1.1 but not IEEE LOM.

The LOM-Editor is another LOM editor written in Java, and developed by Multimedia Communications Lab (KOM) at the Darmstadt University of Technology, Germany (KOM, 2001). The LOM-Editor is written in Java so it needs a Java Runtime Environment (JRE); running the editor with Windows platforms requires at least Version 1.3 of the JRE; Linux/Unix platforms require at least Version 1.2. The LOM Editor supports IEEE LOM Draft Version 6.1. It has a useful GUI, and supports LOM v1.0 vocabulary sets in English, German and Spanish. The editor can be downloaded from the Multibook site (<http://www.multibook.de/lom/en/index.html>).

The RELOAD Editor is a combined content packager and metadata editor that can create, import, edit, and export LOM and content packages. The editor has been developed in scope of a project that focusing on the development of tools that are based on emerging learning technology interoperability specifications. It was funded under the JISC Exchange for Learning Programme (X4L) that is managed by Bolton Institute with staff located at Bolton Institute and the University of Strathclyde (RELOAD, 2004). The RELOAD Editor is a Java application, and should be run on any platform capable of running Java applications. Currently, RELOAD editor can be downloaded from the SourceForge.Net Web site (<http://sourceforge.net/projects/x4l-reload>), and installation packages are available for the Microsoft Windows, Macintosh OSX and Linux plat-

forms. At present, the metadata editor of RELOAD supports IMS Metadata 1.2.2 XML binding but the development team plans to support the IEEE LOM in the future. The users of RELOAD can enter metadata through either form-view or tree view graphical user interface (GUI), interchangeably. However, its tree view editing is powerful its form-view option is limited with one field for each LOM element. The RELOAD is open source software and can be freely modified under GPL for further development by the developers in learning arena.

EM² was developed as an editing environment for educational metadata to provide the basic educational metadata management component of the European IST projects “Knowledge-on-Demand” (Sampson, Papaioannou, & Karadimitriou, 2002). EM² supports both IMS LOM specifications and IEEE LOM including DC. The software was written in Java, and based on XML Document Object Model (XMLDOM) of W3C.

The Customizable Learning Object Metadata Authoring Tool (CLOMAT) has been developed by Malaxa (2003) as a prototype of multi-functional management environment for LOM including SCORM at the Florida State University. It was implemented in ASP.NET and C# and uses the .NET framework connected to a SQL Database Server.

In addition to the tools developed by academic institutions and educational bodies of the local governments, there have recently been intensive efforts of many software vendors for including the LOM to their products, recently. For example, Macromedia uses LO concept, and related standards and specifications with a family of MX products such as FlashMX, DreamweaverMX, Authorware 7, and DirectorMX (Heins & Himes, 2002).

Besides stand-alone and LOM specific editing software, the educational portals or LORs such as MERLOT and CAREO are serving internal tools or integrated components to build LOM by the contributors. Moreover, the *ad hoc* learning systems based on LOM have existed over the last years (Geueke & Stausberg, 2003; Hirata, Takaoka, Ohta, & Ikeda, 2001; Shen, Shi, & Xu, 2002; Steinmann, 2002).

The Learning Object Metadata Generator (LOMGen) can be considered as a leading tool for automatic harvesting and constructing the LOM from the Web. The generator written in Java has been developed by Singh, Boley, & Bhavsar (2004) from the Department of Computer Science of the University of New Brunswick in Canada. The authors reported that the system semi-automatically extracts information from LOs (in HTML files on the Web), constructs key-word/key phrase database, and finally posts them to CANLOM Metadata Repository.

Although many tools and/or systems for application of the metadata standards have recently been developed by some national and international organizations in the USA, Canada, and EU, there is still a strong demand for the tools applying IEEE LOM metadata standard in other domains and languages. Considering the related literature, we can conclude that all of the current tools were mostly developed by using Java technology. However, Java is one of the most popular technologies for open source software whereas dotNet (.NET) has been a rapidly spreading Microsoft framework technology. In fact, at present, except CLOMAT by Malaxa (2003), there is no other work on realization of LOM with .NET found in a Web search done recently in Google search engine. Thus, the pioneering works which will attract a large community of the .NET developers may trigger a rapid increase in number of tools leveraging learning object usage and economy more widely. Since this approach seems to be noteworthy, in this study, the LOM editor has been developed with .NET technology. Thus, we believe that it will contribute to an increase in the number of metadata tools and systems that run under Windows platforms.

XML Binding of LOM for Tree View Editing

LOM standards describe the structure of metadata instances, but they do not define how learning systems will use metadata instances for learning objects. Either “XML binding” or “RDF binding” can be used in order to represent metadata instances of LOM. XML binding of LOM has been applied without problems but finding a simple RDF schema is not easy and many problems still remained unsolved in RDF binding (Bourda & Doan, 2003).

XML binding of LOM can be simply realized in the format of XML. LOM describes the LOs using a set of XML elements divided in 9 main categories such as general, lifecycle, meta-metadata, technical, educational, rights, relation, annotation, classification. Considering the best practice guide published by IMS Global Learning Consortium, “The IEEE conceptual model for metadata definitions is a hierarchy. At the top of the hierarchy is the “root” element containing many sub-elements. If a sub-element itself contains additional sub-elements it is called a “branch”. Sub-elements not containing any sub-element are called “leaves”. This entire hierarchical model is called the “tree structure” of a document” (IMS, 2001). Nilsson, Palmér, & Brase (2003) express that tree structures make it possible to organize the information in the groups formed with related elements.

The tree structure of LOM enables editing of hierarchical metadata with a tree view graphic user interface (GUI) that can be more effectively used by learners, teachers, and librarians. However, many LOM editing tools above reviewed (except RELOAD and CLOMAT) mostly have been developed with GUIs in traditional form view but not tree view. Thus, developing a tree view LOM editor called as “TreeLom” was one of the main objects of this study because of strong editing and navigating advantages of tree view. These advantages are listed below:

- Hierarchical data structure and relationships between elements can be represented and digested more clearly.
- All input fields for the elements can be entirely seen together in a single window. This style speeds up to enter metadata and also reduces the probability of misplacement of metadata.
- Navigation and paging to focus on a particular part of data are easily manageable with expansion and collapsing functionalities when compared to alternative forms of viewing (i.e. tabular form view or list view).

Architecture and Working Model of TreeLom

TreeLom application has been developed with Microsoft’s .NET framework technology. Its code has been written in Microsoft C# but some level of Javascript code also included into GUI to manage navigations at client side. Microsoft .NET Framework is a programming model for the .NET platforms. The key components of the .NET Framework are the common language runtime and the .NET Framework class library including ADO.NET, ASP.NET, and Windows Forms (MSDN, 2004). Generally, the focused component of a tree view LOM editor like TreeLom is a tree view component providing programming functionalities for LOM elements. There are many tree view components available as freeware, shareware, and commercial and they vary in their programming functionalities, peculiarity to certain developing environments and operating platforms. The main component used in TreeLom is the TreeView component, added into MS IEWebControls for ASP.NET. In the new version of ASP.NET (v2.0), an enhanced version of this control is included into framework of ASP.NET and much more powerful than the old one (Walter, 2004). TreeView in IEWebControls is a useful component because of its many methods, properties and events, distributed under Shared Source License for ASP.NET Source Projects. It

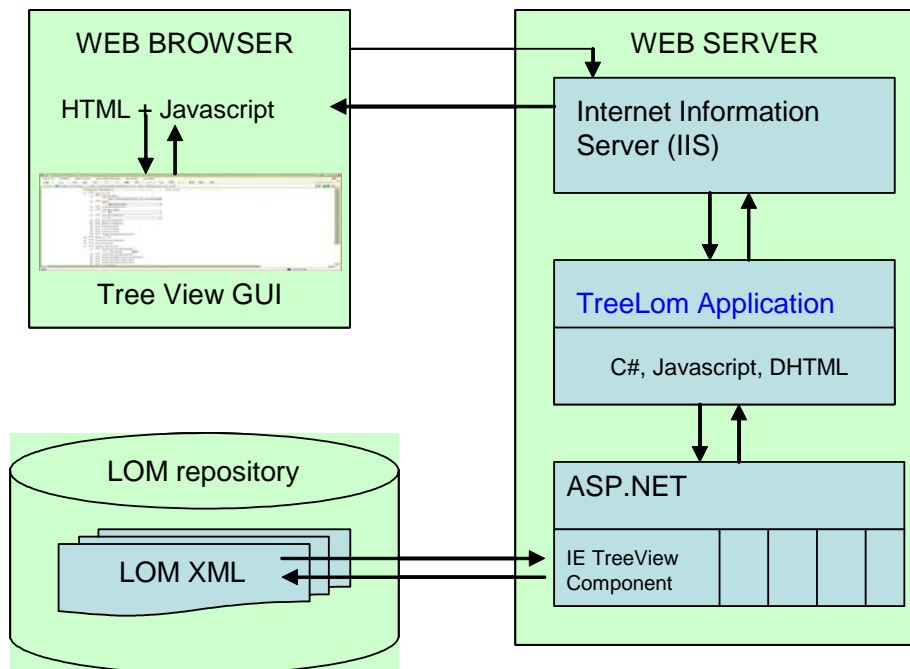


Figure 1: Architecture of TreeLom

supports hierarchical data representation using the ASP.NET server-side control architecture, and also emits DHTML in uplevel browsers (Mitchell, 2004).

The TreeLom editor consists of the server side script files and some support files (the source package can be downloaded from <http://cebeciz.cukurova.edu.tr/tools/treelom>). When a user wishes to access to a Web server hosting TreeLom through its URL, a blank tree view LOM template is displayed in client browser. After editing, when the user clicks “save” button, an XML file is created in a desired path of LOM repository as illustrated in Figure 1. The location of this LOM directory can be replaced in TreeLom source file before installation of the TreeLom package in Web server.

As seen in Figure 2, a simple scrolling menu is used as a support tool containing save, update, collapse, and expand commands on the left side of the GUI of TreeLom. This menu can be replaced with other navigational units if the application is integrated into an existing e-learning system, or it can be recoded for achieving other styles of navigation with DHTML programming.

Even though the current version of TreeView control cannot support the form objects like input and dropdown boxes, this requirement is fulfilled with Javascript programming in development of TreeLom. Therefore, this part of the source code of TreeLom can be considered a good example in combination of Javascript and C# for management of the objects in Web forms. As shown in Figure 2, unlike other tree view LOM editing tools, TreeLom provides a way to enter data in the input boxes or selected from dropdown boxes located just beneath the relevant LOM elements. This process makes the input of metadata much easier and quicker as compared to common model in which all data is entered in a separate textbox located at bottom of the GUI (for example, tree view form of RELOAD). In order to create or delete more instances of a LOM element, the user can use “remove” and “add” command links dynamically displayed next to the related element. In TreeLom, the number and the field length of each LOM element are controlled during operation and before saving process so as to ensure the limitations and definitions made in LOM Draft Standard.

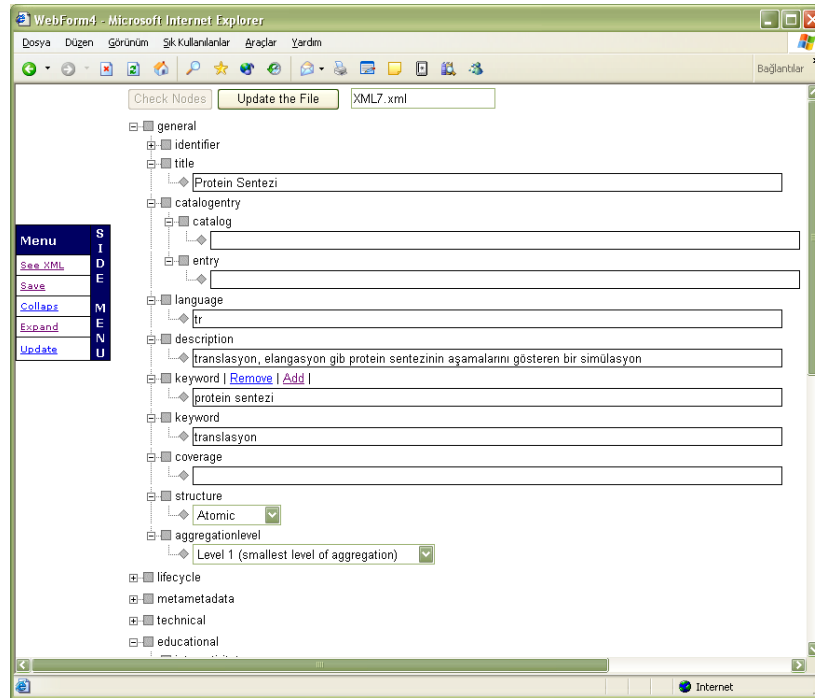


Figure 2: Tree view GUI of TreeLom

In addition, vocabulary management should be an embedded functionality of LOM editors. Ryan and Walmsley (2003) reported that a tool has user-selectable vocabularies for addressing the issue of terminology use and interpretation of terminology. The software would support a number of pre-defined vocabularies the user could select when entering metadata. As Duval and Hodgins (2003) discusses the problems with LOM generation they suggest that metadata can be created, where many of the relevant fields can be prefilled. In design of TreeLom, these suggestions has been taken into consideration with predefined LOM vocabularies to fill in data for some metadata elements such as aggregation level, status, structure, end user role as well as many others. As shown in Figure 2, since the predefined LOM v1.0 vocabularies has been included in TreeLom, users can quickly select the relevant data item from dropdown boxes in TreeLom GUI. This technique reduces the amount of time to enter relevant data for some of LOM elements and it increases the efficiency in data input.

Conclusion and Future Research

Successful implementation of the LOM standard is crucial in order to overcome the complexity of the LOM and to make that learning objects are discoverable, accessible, and reusable. The widespread use of LOM among learning communities depends mainly on availability of powerful tools in the Web. We believe that TreeLom as a freeware tool distributed in the Web will contribute to grow up LOM repositories. Since TreeLom's GUI provides the advantages of tree view editing, LOM instances of learning objects can be easily and quickly created and navigated during data input. Because its tree view GUI lets users hide information items when they do not need their details, then quickly focus on a desired element. Thus, authoring process greatly simplified for inexperienced or new users.

TreeLom can be used as a stand alone application or a subsystem of any LCMS and/or LOR. We hope that TreeLom will especially be useful for the developers who plan to build such kind of systems for .NET platforms because it is just one prototype of LOM editing tools developed in .NET technology. Since the existing LOM editors are mostly stand alone desktop applications, a

great deal of effort is required to adapt them into more complex online systems. On the other hand integrating TreeLom into an online system requires only a minimal installation and modifying experience due to its Web based design architecture.

TreeLom is currently under development. We planned that its core code will be further extended in near future. In future versions of TreeLom we aim to add and develop the following functionalities:

- Options for other metadata specifications and standards such as DC and MARC,
- Conversion functionality for importing and exporting LOM from/to other metadata formats,
- Multilingual LOM vocabularies to support world languages,
- RDF binding option for LOM,
- Validation of the documents for targeted schemas for different versions of LOM.

The success of a learning technology like learning object depends on its widespread use in different applications and platforms. So as we continue to improve and develop interoperability standards for learning we should also try to facilitate its diffusion and acquisition. It is expected that a rapid increase in number of educators using publicly available educational tools including LOM editors will lead a rapid increase in institutional/internal LORs, and in object economy.

References

- Anido, L., Rodríguez, J., Caeiro, M., & Santos, J. M (2003). High-level brokerage services for the e-learning domain. *Computer Standards & Interfaces*, 25(4), 303-327.
- Bourda, Y., & Doan, B-L. (2003). The semantic Web for learning resources. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*. Retrieved on March 24, 2005 from <http://csdl.computer.org/comp/proceedings/icalt/2003/1967/00/19670322.pdf>
- Duval, E., & Hodgins, W. (2003). A LOM research agenda. *Proceedings of 12th International World Wide Web Conference, 24-25 May 2003*, Budapest, Hungary. Retrieved on March 24, 2005 from <http://www2003.org/cdrom/papers/alternate/P659/p659-duval.html.html>
- Duval, E., Hodgins, W., Sutton, S., & Weibel, S. L. (2002). Metadata principles and practicalities. *D-Lib Magazine*, 8(4). Retrieved on March 24, 2005 from <http://www.dlib.org/dlib/april02/weibel/04weibel.html>
- Gueke, M. & Stausberg, J. (2003). A meta-data-based learning resource server for medicine. *Computer Methods and Programs in Biomedicine*. 72(3), 197-208.
- Heins, T. & Himes, F (2002). Macromedia MX: Strategies and architectures for eLearning content. A Macromedia Whitepaper. Retrieved on July 23, 2004 from <http://download.macromedia.com/pub/solutions/downloads/elearning/strategy-architecture.pdf>
- Hirata, K., Takaoka, Y., Ohta, M. & Ikeda, M. (2001). The Meaning of LOM and LOM Authoring tool on HRD. In *Proceedings of the International Conference on Dublin Core and Metadata Applications*, Tokyo, Japan, 2001. Retrieved August 6, 2004 from http://www.ei.sanken.osaka-u.ac.jp/pub/hayashi/DC_Hirata.pdf
- Hodgins, W. (2000). The future of learning objects. In D. A. Wiley (Ed.), *The instructional use of learning objects*. Bloomington, IN: Association for Educational Communications and Technology. Retrieved July 27, 2004 from <http://www.reusability.org/read/chapters/hodgins.doc>
- IEEE LSTC. (2002). IEEE 1484.12.1-2002: Standard for Learning Object Metadata. Retrieved August 8, 2004 from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

- IEEE LTSC. (2004). IEEE P1484.12.3/D2: Draft Standard for Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata. Retrieved August 8, 2004 from http://ltsc.ieee.org/wg12/files/IEEE_1484_12_03_d2.pdf
- IMS. (2001). IMS Learning Resource Meta-data Best Practice and Implementation Guide (Version 1.2 Public Draft Specification). Retrieved January 15, 2004 from http://www.imsproject.org/metadata/ims_md_bestv1p2.html
- KOM. (2001). LOM-Editor Version 1.0, Technische Universität Darmstadt. Retrieved August 20, 2004 from <http://www.multibook.de/lom/en/index.html>
- Lindner, R. (2001). CEN/ISSS/WS-LT Presentation. In *Learning Technology Roundtable Panel Discussion*. March 20, 2001, New York City, USA. Retrieved August 6, 2004 from <http://jtc1sc36.org/doc/36N0085.pdf>
- Malaxa, E. V. (2003). Customizable Learning Object Metadata Authoring. Unpublished M.Sc. Thesis, College Of Arts and Sciences, Florida State University. Retrieved August 10, 2004 from http://etd.lib.fsu.edu/theses/available/etd-08282003-172442/unrestricted/01_evm_thesis.pdf
- MSDN. (2004). NET Framework, Microsoft Developer Network (MSDN) Library. Retrieved August 20, 2004 from <http://msdn.microsoft.com/library/default.asp?url=/downloads/list/netdevframework.asp>
- Mitchell, S. (2004). Using the tree view IE web control. Microsoft Developer Network (MSDN) Library. Retrieved August 20, 2004 from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/aspnet-usingtreeviewwebcontrol.asp>
- Nilsson, M. & Palmér, M. (2002). ImseVimse: The IMS Editor Vimse. Retrieved August 14, 2004 from <http://kmr.nada.kth.se/imsevimse> (Source code from <http://www.sourceforge.net/projects/imsevimse>)
- Nilsson, M., Palmér, M. & Brase, J. (2003). The LOM RDF binding - Principles and implementation. In *Proceedings of the 3rd Annual ARIADNE Conference*, November 20-21, 2003, Leuven, Belgium. Retrieved August 6, 2004 from <http://cid.nada.kth.se/pdf/CID-243.pdf>
- RELOAD. (2004). Reload Metadata and Content Packaging Editor. Retrieved August 20, 2004 from <http://www.reload.ac.uk/editor.html>
- Ryan, B. & Walmsley, S. (2003). Implementing metadata collection: a project's problems and solutions. *IEEE LITF Learning Technology*, 5(1). Retrieved November 20, 2004 from http://ltf.ieee.org/learn_tech/issues/january2003/#3
- Sampson, D., Papaioannou, V. & Karadimitriou, P. (2002). EM²: An environment for editing and management of educational metadata. *Educational Technology & Society*, 5(4). Retrieved August 20, 2004 from http://ifets.ieee.org/periodical/vol_4_2002/sampson_papaioannou.html
- Shen, Z., Shi Y. & Xu, G. (2002). A learning resource metadata management system based on LOM specification. *The Seventh International Conference on CSCW in Design*, September 25-27, 2002, Rio de Janeiro, Brazil. Retrieved July 20, 2004 from <http://www.celtsc.edu.cn/download/shenshixu0209.pdf>
- Singh, A., Boley, H., & Bhavsar, V.C. (2004). A learning object metadata generator applied to computer science terminology. Presentation in *Learning Objects Summit*, March 29-30, 2004, Fredericton, CA. Retrieved July 23, 2004 from <http://www.cs.unb.ca/~boley/treesimilarity/LOMGen-29Mar2004.pdf>
- Steinmann, U. (2002). LOM Editor in the Multimedia Catalogue Version 1.0. Retrieved 6 August, 2004 from <https://mmk.fernuni-hagen.de/lom/Editor>
- Synnytsya, K. (2003). Learning objects metadata: implementations and open issues. *IEEE LITF Learning Technology*, 5(1). Retrieved November 20, 2004 from http://ltf.ieee.org/learn_tech/issues/january2003#0
- Walter, S. (2004, August). Introducing the ASP.NET 2.0 TreeView and menu controls. *MSDN Magazine*. Retrieved August 30, 2004 from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/treeview.asp>

Wiley, D.A. (2003). Learning objects. In A. Kovalchick, & K. Dawson (Eds.), *Educational technology: An encyclopedia*. Santa Barbara: ABC-CLIO. Retrieved July 17, 2004 from <http://wiley.ed.usu.edu/docs/encyc.pdf>

Biographies



Zeynel Cebeci, Ph.D. is a faculty member in the Division of Biometry and Genetics - Faculty of Agriculture at the Cukurova University in Adana, Turkey. In addition, he has been the director of the Computer Sciences Research and Application Center at the Cukurova University from 2000 to 2004. He currently teaches courses on biometrical methods and information/communication technologies. His research interests include programming languages, software development, applications on genetics (computational biology), biometrical methods and algorithms. He is currently conducting research and implementation works on e-learning systems and technology.



Yoldas Erdogan is an instructor in the Department of Informatics at the Cukurova University in Adana, Turkey. He teaches courses on programming languages at the Department of Informatics. In addition, he is the principal developer of "e-Cukurova Learning Content Management System", used in distance education programs of the Cukurova University. His research interests are programming with .NET technology, scripting languages and Web design.