8

Designing and Testing an Open-Source Learning Management System for Small-Scale Users

KEVIN JOHNSON AND TIMOTHY HALL

Abstract. The vision of reusable learning resources or objects, made accessible through coordinated repository architectures and metadata structures, has gained considerable attention within education and training communities. A proliferation of standards, architectures, Web technologies, and functionality abound to help realize this promise. This chapter outlines the issues associated with designing solutions for small-scale users such as small to medium-sized enterprises (SMEs). It describes the requirements and architecture for the development of an open-source small-scale learning object (LO) management system that supports the full management of learning objects, by bringing together the most promising advances in this field to attain a learning system for use by small-scale users to leverage the power of learning objects for improved training at an individual and organisational level.

8.1 Introduction

This chapter focuses on a solution to the problem of the diverse and changing information, training, and learning needs associated with small-scale systems that are needed by small to medium-sized enterprises (SME) and the mismatch between current e-learning systems and content-generation techniques, and the needs of their customers and employees. We can extrapolate the solution that matches this SME need to the promised more personalized e-learning resources of the future.

E-learning initiatives are frequently driven by an awareness of knowledge as an important source of wealth creation and a need to respond to the quickening pace of environmental change and the rapid development of information technologies. However, various criticisms have been raised about the marginal benefits that arise from using technology in education. Whether or not there are advantages to taking this route still remains to be seen. Despite these concerns, learning technology is continually advancing, and new initiatives are underway to standardize e-learning tools, technologies, and content [1].

Attempting to take advantage of developments in learning technologies, companies have purchased off-the-shelf e-learning courses, only to find them a less than satisfactory solution to their training needs. These courses are too generic and cannot economically be tailored for a better match to specific company needs. However, recent advances in learning standards (such as SCORM [2], IEEE LOM [3], and IMS [4]) and in using XML (eXtensible Markup Language) to classify content now make it possible to create learning content management systems (LCMSs), that can handle the required diversity and specificity efficiently. An LCMS is an environment where developers can create, store, reuse, manage, and deliver learning content, based on a learning object model, from a central object repository (database), with good search capabilities to find the text or other media needed to build training content quickly. LCMS strives to achieve a separation of content, tagged in XML, from a Web browser–based presentation framework; this facilitates publishing a wide range of formats, platforms, or devices, all from the same source content material.

These content fragments have become known as learning objects. A learning object, for all practical purposes, is an object or set of resources that can be used for facilitating intended learning outcomes and can be extracted and reused in other learning environments. The term has been recently associated with electronic learning resources that can be shared in multiple learning environments. The value of a learning object lies in its object-oriented nature, which lends itself to reuse; however, therein lies its complexity. Two major issues that affect the pedagogical validity of a learning object are granularity and combination. Stated succinctly, combination relates to how the learning objects are amalgamated, whereas granularity refers to the size of the learning object itself.

Current-generation commercial LCMSs for the education market—WebCT [5], Blackboard [6], TopClass [7]—and for the corporate training market—Docent [8] and TrainerSoft [9]—offer a "corporate solution" and include functions such as mail services, authentication services, intranets, or the Internet. These are services that many users already have, so duplication would be a waste of resources. Small companies are additionally limited by the capital required for purchasing such expensive proprietary systems. From the system engineer's point of view, these products are not easily modified to include tailored features. Commercial LCMSs, such as WebCT and Blackboard, however, have valuable features that are not easy to duplicate, for example, tools to support the creation of instructionally sound learning content that normally would require the services of a trained instructional designer.

Our task was to create a small-scale LCMS environment that met the needs of SMEs (as representative of a class of similar small-scale users), countering their neglect in the current drive toward the use of e-learning. The proposed solution would have to develop a methodology and technology that would address the inhibiting issues of cost and complexity, and allow SMEs to take advantage of e-learning for skills development and for use in training/educating its clients, representatives, field support personnel, etc., and incorporating elements of sound instructional design for use by those without training design experience. By basing the system on reusable learning objects, the customization of modules of learning tailored to suit the individual SMEs learning requirements was possible, and a flexible dynamic delivery system provided a “just-in-time” and “just-enough” learning approach.

By following our development path, we explore the associated issues and problems and illustrate the system design process in a time of shifting and developing standards.

8.2 Learning Management Systems to Learning Content Management Systems

Although it is easy enough to provide access to a piece of learning content directly from a Web page, many organizations and educational institutions want to control access to the courseware and track data such as the user ID, the level of usage, and the outcome. A learning management system (LMS) is a Web server–based software application that provides the administrative and data-tracking functions necessary to achieve this. The LMS also relieves the teacher of a burdensome administrative effort. The specific features and functions of an LMS vary considerably from one system to another, but generally they offer the following:

- Administrative functions: these include course setup, learner registration, course assignment and reporting of the learner’s progress by tracking data such as scores on tests or quizzes, the time spent in courses, and the completion status of each course.
- Learner interface: permitting learners to log in to the LMS using a personal ID with or without a password and receive access to the e-learning content via a personalized menu of their assigned courses. Usually they can also monitor their own progress by viewing test scores, completion status on courses and topics, and so on.
- Sequencing: LMSs are also responsible for sequencing learner access to lessons within courses, such as allowing learners access lessons in any order or forcing them to access the lessons in a predetermined sequence.

An LMS enables organizations to collect data about the level of usage and effectiveness of courses. Usage data includes the number of learners registered for a course, the average amount of time spent on a course, and the number of learners completing a course.

Learning content management systems (LCMSs) are a more recent development that exploits the wider use of standards-compliant learning objects to add a further level of functionality to LMSs. However, a natural result of the adoption of learning objects is that there is a much larger number of content pieces to deal with; thus LCMSs, need more advanced content management, organization, and search capabilities. The systems are designed to meet the following requirements [10]:

- Generate unique descriptions for each learning object.
- Discover (search for and locate) the required learning object.

- Provide multiple hierarchies for storing and organizing learning objects.
- Facilitate the assembly of complex course structures.

A typical LCMS includes the following components:

- Content tagging and assembly functions for creating learning objects from lower-level content objects and for grouping learning objects to form larger learning content structures such as topics, lessons, and courses.
- A content repository for storing assets, learning objects, content aggregations and other content structure.
- A delivery interface including functions for searching and organizing learning objects to provide individualized learning experiences.
- Authoring tools for producing content objects.
- Some form of collaboration tool that allows the end users to talk among themselves as well as post questions and queries to someone who administers the course. This tool can be in the form of a chat room or instant message system (synchronous) or a forum or bulletin board area (asynchronous).

The international research consultants at IDC (2002), in their paper "Learning Content Management Systems: A New E-Learning Market Segment Emerges," identify the components of an LCMS as consisting of an authoring application, a data repository, a delivery interface, and administration tools. Many other vendors such as Click2Learn and Avaltus [11] also concur with this architectural structure. The authoring tools provide templates and storyboarding capabilities, and can be used to convert existing content. The data repository uses metadata to store and manage individual learning objects. The delivery interface dynamically delivers content that can be modified to reflect the required look or feel.

Most LCMSs additionally contain LMS functionality with administration functions to manage learner profiles, assessment, and course catalogues, and provide a learner interface.

8.3 Reusability and Interoperability

Ultimately, the usefulness of these environments from a teaching and learning point of view is their ability to assemble and deliver lessons and courses from granular pieces of instruction based on learning objects, and their ability to take an object and reuse it in a different lesson (reusability) or modify it for a different definition of repurpose; from a technical point of view, a learning object should be deliverable through a different LCMS than that in which it authored (interoperability). This is a very different concept than the majority of existing e-learning, where courses are one indivisible unit delivered only through the original system.

8.3.1 Reusability

Reusability has more than one meaning when associated with e-learning and learning content management systems. For now we'll focus on content reusability, that is, the reusability of the material delivered to the learner to achieve a learning goal.

- ICT-based delivery has several advantageous features that made its adoption as a means of industrial training, that is computer-based training (CBT) attractive.
- Media-rich interactive CBT was a far more effective training tool that printed manuals.
- The addition of assessment and data tracking meant that management could be assured that its personnel met the required standards.
- CBT was available to personnel 24 hours per day, 7 days per week (24/7).
- Personnel could access the training material on a just-in-time basis, so that they could carry out a particular task immediately after reviewing the latest information [10].

The CBT was delivered stand-alone, on CD-ROMs, or across a local area network for multiple user access. However, authoring this rich content is time-consuming, 100 times the delivery time not being uncommon. CBT systems had delivery programs designed for specific end users, and reuse of the content for other purposes was not seen as an important factor [12]. Early e-learning content followed this path, with significant resources being devoted to authoring locked-in content.

With the spread of e-learning to other less well resourced areas of education and training, much of the research into the creation of learning content has focused on authoring resource economy, and the notion of reusable rich media content components and learning objects becomes attractive. The driving force is that reuse of such components can lead to important savings in time and money, whereas richer media enhances the quality of the learning experience. The end result is faster, cheaper, more effective learning.

Reuse of learning content is not simple. It comes in a number of conceptually or technically different guises, for example:

- Multiple output (distribution) formats, or media
- Multiple purposes: training, performance support, reference documentation, marketing information, etc.
- Multiple delivery: the same material over and over
- Multiple "disciplines" or market segments

Reuse does not involve any change in the learning content, but if we extend the principle of resource conservation to allow a reuse that involves a degree of modification or reauthoring of learning objects (LOs), we arrive at the concept of repurposing, which can be thought of as the ability to use, without any (significant) changes, the same piece of content for a purpose significantly different from what it was originally intended for when created [13]. We do not pursue this topic further here.

Again extending the meaning of reuse along a technical route, we must consider enabling content to be delivered through other systems than that in which it originated; this is termed interoperability.

8.3.2 Interoperability

Interoperability is defined as “enabling information that originates in one context to be used in another in ways that are as highly automated as possible” [14]. More specifically: the ability of objects from different, multiple, potentially unknown or unplanned sources to “work” or operate when put together with other objects. Examples include:

- Content objects from different original creation/authoring tools working together when assembled into a learning object.
- Learning objects and content objects being able to work properly when moved from one infrastructure (operating system, LCMS, etc.) to another

This requires standardization of common protocols, formats, etc. The vision of an open, large-scale learning object infrastructure is conditional on the achievement of interoperability.

Interoperability can exist at different scales:

- Between learning objects
- Between learning objects and learning management systems
- Between learning object repositories
- Between metadata schemas

The more general notion of interoperability is that it enables crossing cultural or linguistic boundaries. Interoperability requires full exchange of data between the systems’ heterogeneous data models. For an exchange to take place, a consistent set of interpretations must be provided for the information. Ensuring this consistency requires semantic interoperability, in other words, agreement on the meaning of the exchanged information [15]. Accordingly, “the achievement of interoperability should be viewed as an enabling condition for interoperation between application systems and semantic integration of information from diverse sources” [16]. Thus, interoperability relies heavily upon communication of information between systems, applications, and databases wherein formal language and model representations of complex information have been resolved.

Efforts to create standards for the interchange of information or metadata over the past 10 to 15 years have produced a number of national and international standards. The prevalent approach has been to develop interfaces that allow translation of data from one proprietary format to a standard or “neutral” format, from which the information can again be translated into a second proprietary format. Much effort has been directed at formalizing general aspects of storing and retrieving properties and entities, most notably by the IMS [4], IEEE [17], AICC [18], and ARIADNE [19].

Metadata comprise a key component of any interoperability schema. As the format of metadata evolves toward machine readability, improved reliability and consistency in the interchange of information occurs. Further work is needed in storing and representing metadata, specifying metadata requirements for different domains, and building tools that are able to find commonalties between interchanged data from different agencies [20].

8.4 Metadata

Metadata are often defined as “data about data” [21] and are understood to represent descriptive information (element names, definitions, lengths, etc.) about populated data fields. Benefits of implementing a metadata model are seen in:

- Locating information: metadata associate information with objects that otherwise would not exist or are not easily accessible. This in turn benefits searching for a specific object and returns a higher percentage of accurate results.
- Interpreting information: metadata fields associated with objects offer a clearer description of an object and better define what the object is about [22].

Metadata support the search for information by providing data definitions, transformation logic, and lists of valid values, business rules, and more. The main components within a metadata system are the repository that holds all of the information, the user interface, and the interface to other software and publishing, both electronic and paper [23].

The repository captures the metadata, usually in a relational database. All repositories hold the basics: length, definition, data type, etc., and additionally, source and target mappings, the relationship between elements, and much more.

The user interface allows the metadata administrator to enter and maintain records, though most entries come to the system through data uploads or interface with other software. Metadata maintenance can be surprisingly complex, so an intuitive and powerful user interface is important.

The software interfaces both receive and send information about the data to any applications that may touch or define data, such as a data modeling tool, business modeling tool, RDBMS (Relational DataBase Management System), change management tools, and testing support tools. At the moment, this is a strong developmental area of metadata systems [24].

Publishing makes metadata available to the business and technical user community. Usually published metadata are viewable via an Internet browser window and on hardcopy reports such as mapping specifications or a data dictionary. Not all information captured in the repository is publishable, and the amount of control over the user interface and report designs varies among metadata tools.

In order for the positive potential of learning objects to be realized, they need to be labeled, described, investigated, and understood in ways that make the simplicity, compatibility, and advantages claimed for them readily apparent to teachers, trainers, and other practitioners [22]. The information to enable this must be stored in the associated metadata.

Standards—whether they are for data collection, data transfer, documentation (metadata), or software—are all designed to facilitate the dissemination, communication, and use of information by multiple producers and users. (Almost all standards rely on or incorporate metadata in order to accomplish their purpose.)

Recent trends in education are also highlighting the importance of metadata, as the vast amount of educational material on the Web needs to be cataloged and organized in a standardized way so that it can be utilized interoperably for different educational environments [4].

We have established a framework for e-learning content to be assembled for delivery dynamically from a repository, where the pieces are located and sequenced according to the metadata, but what about the pieces of learning themselves, the learning objects?

8.5 Learning Objects (LOs)

Technology is an agent of change, and major technological innovations can result in entire paradigm shifts. The computer network known as the Internet is one such innovation. After effecting sweeping changes in the way people communicate and do business, the Internet has begun to bring about a paradigm shift in the way people learn. Consequently, a major change may also be coming in the way educational materials are designed, developed, and delivered to those who wish to learn. An instructional technology called “learning objects” [25] currently leads other candidates for the position of technology of choice in the next generation of instructional design, development, and delivery, due to its potential for reusability, generativity, adaptability, and scalability [26,27].

Learning objects, as discussed in Chapter 1, are elements of computer-based instruction grounded in the object-oriented concept. Object-orientation highly values the creation of components (called “objects”) that can be reused [28] in multiple contexts. This is the fundamental idea behind learning objects; instructional designers can build small (relative to the size of an entire course) instructional components that can be reused in different learning contexts. Learning objects are generally understood to be digital entities deliverable over the Internet. Any number of people can access and use them simultaneously (as opposed to traditional instructional media, a book, or video tape, which can only exist in one place at a time).

Supporting the notion of small, reusable pieces of instructional media, Reigeluth and Nelson [29] suggest that when teachers first gain access to new material, they often break it down into constituent parts. They then reassemble these parts in ways that support their individual instructional goals. This suggests one reason why reusable instructional components—learning objects—may provide significant benefits. If instructors had access resources as components in the first place, the initial step of decomposition could be bypassed, increasing the speed and efficiency of instructional development.

The IEEE Learning Technology Standards Committee chose the term *learning objects* to describe these small instructional components, established a working group, and provided a working definition.

Various other terms are in use including *content object, knowledge object*, *reusable information object*, and *reusable learning object*. Although no universal definition exists, a learning object generally refers to a "reusable unit of learning." An initial definition for a learning object could be any entity, digital or nondigital, that can be used, reused, or referenced during technology-supported learning. Examples of technology-supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of learning objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organisations, or events referenced during technology-supported learning [30].

This definition is extremely broad—too broad. It failed to exclude any person, place, thing, or idea that had existed at anytime, ever, since any of these could be "referenced during technology supported learning." Different groups have attempted to narrow the scope of this canonical definition to something more specific. Other groups had refined the definition but continued to use the term *learning object*. Confusingly, these additional terms and differently defined learning objects are all Learning Technology Standards Committee learning objects in the strictest sense. The proliferation of definitions for the term *learning object* has made communication confusing and difficult.

The Learning Technology Standards Committee definition seems too broad to be useful, since most instructional technologists would not consider the historical event the First World War or the historical figure Billy the Kid to be learning objects. At the same time, the creation of yet another term only seemed to add to the confusion, so in the context of this chapter, a learning object is defined as "any digital resource that can be reused to support learning." This definition includes anything that can be delivered across the network on demand, be it large or small. Examples of smaller reusable digital resources include digital images or photos, live data feeds, live or prerecorded video or audio snippets, bits of text, animations, and smaller Web-delivered applications, such as a Java calculator. Examples of larger reusable digital resources include entire Web pages that combine text, images, and other media or applications to deliver complete experiences, such as a complete instructional event. This definition of learning object, "any digital resource that can be reused to support learning," is used for two reasons. First, it is sufficiently narrow to define a reasonably homogeneous set of things: reusable digital resources. At the same time, the definition is broad enough to include the estimated 15 terabytes of information available on the publicly accessible Internet [31]. Second, it is based on the LTSC definition (and defines a proper subset of learning objects as defined by the LTSC), making issues of compatibility of learning object and learning object as defined by the LTSC explicit. It captures the critical attributes of a learning object, "reusable," "digital," "resource," and "learning," but rejects aspects of the LTSC that include nondigital and nonlearning focused.

A learning object is thus, for all practical purposes, an object or set of resources that can be used to facilitate intended learning outcomes and can be extracted and reused in other learning environments, "reusable learning objects" (RLOs). Learning objects become the building blocks of e-learning content and can be used to construct any desired type of learning experience—Legos for e-learning [32,33].

Many educators see learning objects as a viable alternative to the traditional yet not very flexible and difficult-to-adapt instructor-led course format that has been the foundation of education and training for the last two centuries. Learning objects stored in a database and properly tagged for easy search are designed specifically for flexibility and reuse and are easily aggregated into lessons and courses.

The value of a learning object lies in its object-orientated nature, which lends itself to reuse. However, therein lies its complexity. Two major issues that affect the pedagogic validity of a learning object are granularity and combination—combination relating to how the learning objects are amalgamated, and granularity referring to the size of the learning object.

8.5.1 Combination

While groups like the Learning Technology Standards Committee exist to promote international discussion about the standards necessary to support learning object–based instruction, apparently no one had considered the role of instructional design in composing and personalizing lessons [34]. Metadata, descriptive information about a resource such as title, author, version, format, etc., facilitate finding objects by searching, as opposed to browsing. Problems arose when consideration was given to what it means for a computer to automatically and dynamically assemble a lesson, by taking individual learning objects and combining them in a way that makes sense: in instructional design terms, "sequencing" the learning objects. In order for a computer to make sequencing or any other instructional design decisions, it must have access to instructional design information to support the decision-making process. However, no such information was included in the metadata specified by the version of the Learning Objects Metadata Working Group standard in use at the time. An IEEE LOM working group is considering this problem [21].

8.5.2 Granularity

Sequencing cannot be discussed without mentioning "granularity" [35]. How big should a learning object be? The Learning Technology Standards Committee's definition leaves room for an entire curriculum to be viewed as a learning object, but objects so large preclude notions of reuse that lies at the core of learning object features, as generativity, adaptivity, and other-ivities are all facilitated by the property of reuse. Clearly LOs should be smaller and from a reuse point of view as small as possible. Unfortunately, it's not so straightforward. Learning objects must be tagged with metadata (with more than 20 fields with names like "Semantic Density"), very small objects become prohibitively expensive to tag, a trade-off between flexibility of reuse and the cost of tagging has to be made, and an intermediate size for LOs chosen. Alternatively, the decision between how much or how little to include in learning objects can be viewed as a problem of "scope." Reality dictates that cost must be considered, but only after decisions regarding the scope of learning objects have be made in an instructionally grounded, principled manner.

To facilitate the ability to find and share learning object's, various standards groups have worked together to define a consistent set of metadata to be provided for each learning object. The metadata is not part of the learning object itself; rather, it is held in a separate document designed to travel with the learning object, and this document is accessed without opening or displaying the actual LO content.

As described earlier, LOs can be considered the building blocks of e-learning content. Building blocks are not particularly useful unless they are assembled into larger structures. Most learning content, regardless of how it is delivered, uses some sort of hierarchical structure. A course may be divided into lessons, for example, and the lessons further divided into topics, and so on. A major requirement for e-learning specifications is the provision of a simple but flexible method for representing a wide variety of content structures or taxonomies.

8.6 Standards

National and international committees, consortia, and other organizations have been busy developing standards and specifications for e-learning technologies at least since the late 1990s. They have been doing so with the understanding that the benefits of this standardization work will be manifold and various:

> Not only would the development and use of international standards (in e-learning) produce a direct cost savings, but the information technology systems could be used in a wider range of applications, and used more efficiently. Better, more efficient and interoperable systems, content, and components will produce better learning, education, and training—which has a positive effect upon all societies [36].

Organizations actively developing these standards and specifications include the IMS Global E-Learning Consortium, the IEEE Learning Technologies Standards Committee, and the ISO Subcommittee on "Information Technology for Learning Education and Training." The development of technical standards in e-learning can be understood as a part of the maturation of this sector or industry. Before, and especially since, the popular emergence of the Internet and the World Wide Web, ICT has been used widely in education, both distance and classroom based, and in off-line and online training. However, the technology has been applied in ad hoc and diverse forms, innumerable courses, course components; and systems for managing and delivering these courses have been developed independently of one another. Moreover, the content and management systems are often created in a manner that makes it very difficult if not impossible to enable content sharing or successful interoperation. Standards in e-learning seek to address these shortcomings by ensuring the interoperability, portability, and reusability of content and compatibility of systems. Until the emergence of standards in the e-learning industry, organizations were often constrained to buying all their e-learning from one vendor. Courses came with their complete software already integrated, and although data flowed freely between the LMS and the courseware, there was no way that courses or LMS could interoperate with another vendor's system. Customers were effectively locked into one vendor.

The observation that "the nice thing about standards is that there are so many to choose from" [22] has been circulating in e-learning standards circles for some time. Although no one involved in standards development would claim to be seeking a situation in which standards and specifications compete, overlap, or develop in parallel, this statement certainly reflects the varied and complex nature of standards organizations and standards development processes.

Standards can be defined as "documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose" [36]. In e-learning the standards that are in use today are a result of the work of several standards bodies, principally the Institute of Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC) [25], the IMS Global Consortium [17], Advanced Distributed Learning Networks (ADL Net) [37], and the Aviation Industry Computer Based Training Committee (AICC) [18], and they ultimately define the metadata to be used in tagging LOs. Eventually the international organization will advance most of the standards developed by the IEEE/LTSC as International Standards for Standardization (ISO).

8.6.1 Standards Evolution

The IMS project was founded as part of the National Learning Infrastructure Initiative of EDUCAUSE (then Educom) as a fee-based consortium of learning-technology vendors, publishers, and users. Its members included many U.S. universities, and its original focus was on higher education. It produced specifications covering multiple areas of e-learning—metadata, content, administrative systems, and learner information—each developed by its own working group. IMS later relaunched as a nonprofit organization with a more international outlook, the IMS (Instructional Management System) Global Learning Consortium [38].

IMS produces open specifications for locating and using e-learning content, tracking learner progress, reporting learner performance, and exchanging student records between administrative systems such as LMSs. Two of these specifications have been adapted for use within the ADL framework:

- The IMS Learning Resources Metadata Specification defines a method for describing learning resources so that they can be located using metadata search software.
- The IMS Content and Packaging Specification defines how to create reusable learning objects that can be accessed by a variety of administration systems such as LMSs and LCMSs.

The Open University of the Netherlands (OUNL) was the creator of EML (Educational Modeling Language) over a 3-year R&D program and was closely involved in the development of the learning design specification in IMS [39]. Currently they are collaborating with the dotLRN community to integrate their learning platform with instructional design defined according to the current standards.

The ADL (Advanced Distributed Learning) common technical framework is referred to as SCORM—the Sharable Content Object Reference Model (SCORM^{TM}). SCORM defines a Web-based learning Content Aggregation Model and Run-time Environment for learning objects [37]. At its simplest, it is a model that references a set of interrelated technical specifications and guidelines designed to meet the Department of Defense's high-level requirements for Web-based learning content. The SCORM applies current technology developments—from groups such as the IMS Global Learning Consortium, Inc., the Aviation Industry CBT Committee, the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) [19], and the IEEE LTSC—to a specific content model to produce recommendations for consistent implementations by the vendor community.

SCORM is being developed through active collaboration among private industry, education, and the U.S. federal government with the goal of producing guidelines that meet the common needs of all sectors. To facilitate this collaboration, the ADL established the ADL Co-Laboratory Network, which provides an open collaborative environment for sharing and testing learning technology research, development, and assessments [40]. Rather than reinventing the wheel, the SCORM leverages the work of the standards bodies by bringing together their disparate specifications and adapting them to form an integrated and cohesive implementation model.

SCORM documents are constantly evolving as further specifications are refined and added to the base model. Figure 8.1 gives an overview of the SCORM structure in its book format bases on the SCORM 2004 documentation.

The AICC develops technical guidelines known as AICC Guidelines and Recommendations (AGRs). An AGR is a short document that references a detailed specification document. AGR 010 is the AICC's guidelines for interoperability between Web-based courseware and LMSs. It references another document, CMI001—"CMI Guidelines for Interoperability"—which is commonly referred to in the e-learning industry as the AICC CMI specification.

The AICC offers certification testing for the AGR 010 CMI interoperability guidelines as well as for the AGR 006 guidelines, which apply to LAN-based management systems. To achieve AICC certification, products are put through a testing process by an independent third-party testing organization. Vendors are also able to self-test their products using the AICC test suite. This enables them to claim AICC conformance for their products.

The ARIADNE European Projects (phases I and II) were formed to develop a set of e-learning tools and methodologies. The ARIADNE began research and technology development projects in January 1996. These projects pertain to the Telematics for Education and Training sector of the 4th Framework Program for R&D of the European Union. The projects focus on the development of tools and

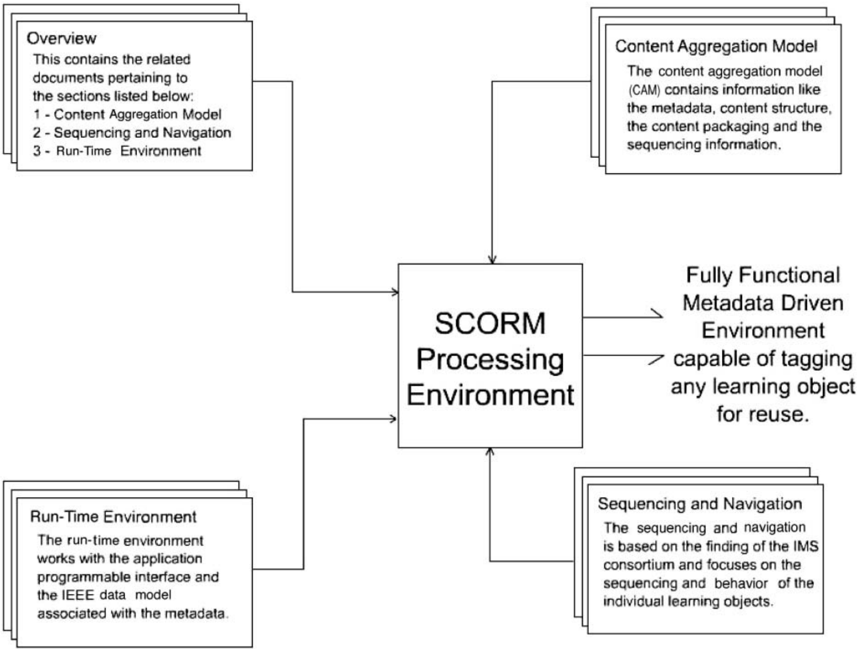


FIGURE 8.1. SCORM "books" outlook.

methodologies for producing, managing, and reusing computer-based pedagogical elements and telematics supported training curricula. The project, which was largely funded by the European Union and the Swiss government, ended in June 2000. Subsequently, the ARIADNE Foundation formed to promote the widespread adoption of state-of-the-art and platform-independent education in Europe [41].

The International Standards Organization (ISO) is a worldwide federation of national standards bodies from some 140 countries [42]. It has created a Joint Technical Committee in cooperation with the International Electrotechnical Commission (IEC), which is the international standards and conformity assessment body for all fields of electrotechnology [43]. This technical committee, known as JTC1, includes a subcommittee known as SC36 (subcommittee 36), which is responsible for work on information technology for learning, education, and training [44].

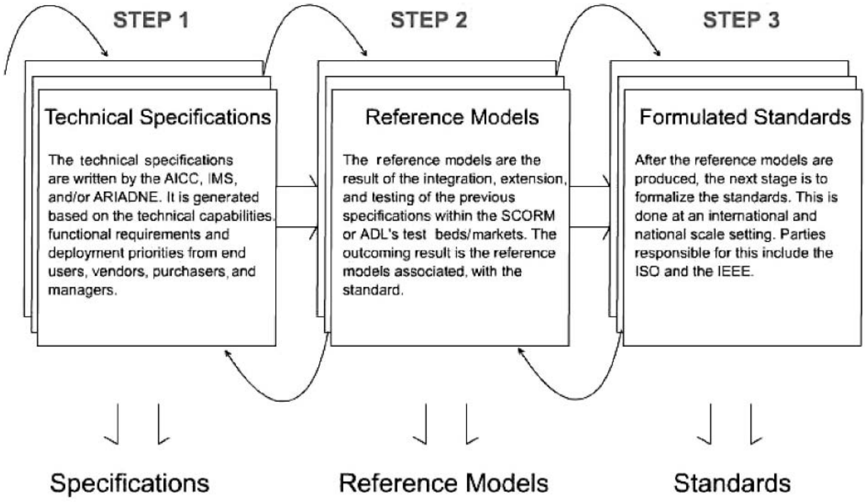
The bodies and organizations listed previously have been working together to create a specification or standard that would allow all users of learning object-based content to define interoperable metadata for learning objects, a standard known as the Learning Object Metadata standard.

To facilitate the widespread adoption of the learning object approach, the IEEE LTSC formed in 1996 to develop and promote instructional technology standards [25]. Without such standards, universities, corporations, and other organizations around the world would have no way of ensuring the interoperability of their instructional technologies, specifically their learning objects. Multiple organizations (ADL, AICC, IMS, ARIADNE) began developing technical standards to support the broad deployment of learning objects. Many of these local standards efforts have representatives on the LTSC group.

8.6.2 Learning Object Metadata Standards

An emerging standard is developing for learning objects metadata. The IEEE is the main accredited standards body (Fig. 8.2). The approved IEEE Learning Object Metadata standard is created with the cooperation of a variety of specification consortia and laboratory test beds and markets. Technical specifications are developed by the AICC, IMS, and ARIADNE, and feed into reference models for ADL [2] and ALIC [45]. These reference models, in turn, aid the standard bodies in developing approved standards. Each of these organizations and their role are outlined below.

The IMS gathers functional requirements, technical capabilities and deployment priorities from end users, vendor, purchasers, and managers. These requirements are consolidated into one or more specifications. These specifications have become a draft for the Learning Objects Metadata specification of the IEEE standards body. The active groups with IMS follow an open process to develop a specification package consisting not only of content metadata but also information models, XML



The technical specifications are written by the AICC, IMS, and/or ARIADNE. It is generated based on the technical capabilities, functional requirements and deployment priorities from end users, vendors, purchasers, and managers.

The reference models are the result of the integration, extension, and testing of the previous specifications within the SCORM or ADL's test beds/markets. The outcoming result is the reference models associated, with the standard.

After the reference models are produced, the next stage is to formalize the standards. This is done at an international and national scale setting. Parties responsible for this include the ISO and the IEEE.

Note: The creation of an affiliated standard is a long and slow process. The steps listed above are repeated over and over until a satisfactory outcome is achieved. The specifications stage is iterated until all contributing bodies are happy with the proposed outcome. This is, in turn, passed on to the reference model test beds. Here it will be tested and retested based on features such as implementation guidelines, requirements etc. When it has sucessfully passed this stage, the ISO and IEEE formally standardize it.

FIGURE 8.2. Standards development process.

binding, and best practice guides. Similar efforts started in the ARIADNE project groups on metadata definitions, and these groups are now closely collaborating with IMS.

In addition, the AICC has been working to provide interoperability standards for computer-managed instruction systems, now more widely known as Learning Management Systems or Course Management Systems (CMS). AICC primarily caters to the CMI systems developed for the aviation industry and related vendors. It provides AICC guidelines and recommendations. The contribution of AICC is particularly important in the CMI database schema and the interoperability of the database objects extending to several computer-based training course management and assessment systems. AICC is working closely with the IEEE LTSC in several areas of mutual interest. It also provides test suites for AICC certification.

Following on from the technical specifications produced, reference models are extrapolated. One organization involved in efforts toward that end is the ADL. The activities of ADL co-labs focus on the development of the Sharable Content Object Reference Model specifications, including metadata standards from IEEE LTSC P1484 (as submitted by IMS) and CMI database schema (submitted to IEEE by AICC).

The intent of ADL co-labs in the development of standards is to make SCORM an integrated model reliant on extended specifications from other groups. ADL participates with other organizations, such as AICC and IMS, in the development of specifications, and when the specifications become stable, it incorporates them into a SCORM release.

The ADL Co-Labs are collaborating closely with ARIADNE, IMS, AICC, and IEEE. At present, SCORM 2004 is distributed and includes the content packaging and sequencing recommendations proposed by IMS. The specific goal for SCORM is to create learning technology standards for the creation of durable, reusable, interoperable, and accessible courses for defense and industry training.

Partners such as ARIADNE, IMS, and AICC have recognized that it would be inappropriate to develop competing metadata systems. They therefore have agreed to cooperate under the auspices of the IEEE LTSC. It is hoped that this will lead to a joint adoption of the metadata standards while retaining the option of producing extensions to these standards that address the particular needs of the respective projects. All four organizations have participated in the development of the IEEE LTSC standards. The procedure for this development of standards is as follows:

- Technical specifications are written within AICC, IMS, or ARIADNE.
- They are integrated, extended, and tested in SCORM/ADL generating reference models.
- They are formalized nationally and internationally in ISO/IEEE.

From this a formal ISO standard is created. The Learning Object Model (LOM), standard was approved by the IEEE in July 2002. It represents the first standard for learning content to be released by an accredited standards organization. The official designation of the LOM standard is IEEE 1484.12.1-2002 [25]. According to the LOM,

> The purpose of this standard is to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors. The purpose is also to facilitate the sharing and exchange of learning objects, by enabling the development of catalogues and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata will be exploited [25].

Currently, the IEEE LTSC learning object metadata standard specifies the syntax and semantics of learning object metadata, to fully/adequately describe a learning object. It focuses on the minimal set of attributes needed to allow learning objects to be managed, located, and evaluated. The standards accommodate the ability of locally extending the basic fields and entity types, and the fields can have a status of obligatory (must be present) or optional (may be absent). Relevant attributes of learning objects that can be described include type of object, author, owner, terms of distribution, and format. Where applicable, learning object metadata may also include pedagogical attributes such as teaching or interaction style, grade level, mastery level, and prerequisites. It is possible for any given learning object to have more than one set of learning object metadata. The standard does not concern itself with how these features are implemented.

The IEEE LTSC LOM model has nine categories, and each category is broken down into constituent parts that further describe a learning object. The categories are:

- General—information describing the LO as a whole
- Life cycle—contains information about the life cycle and status of the LO
- Meta-metadata—information about the metadata that describes the LO
- Technical—technical requirements and characteristics of the LO
- Educational—information about the interactivity type and selected difficulty of the LO and any pedagogical details
- Rights—information about the copyright issues associated with the LO
- Relation—relative LOs in a similar area
- Annotation—history of who created the LO and when
- Classification—where the LO falls within a particular classification system

8.7 Learning Object Metadata (LOM)

The simplest definition of metadata is structured “data about data.” Metadata is defined as “something that describes an information resource, or helps provide access to an information resource” [34]. Metadata are descriptive information about an object or resource whether it is physical or electronic.

While metadata itself is relatively new, the underlying concepts have been in use for as long as collections of information have been organized. Inherent is the concept of an association between the metadata and the information resource that it describes. For example, library card catalogues represent a well-established type of metadata that have served as collection management and resource discovery tools for decades. Metadata can be generated either "by hand" or derived automatically using software.

Metadata standards are applied by using a "template" that, upon completion, holds sufficient information about the object or learning material for a search of the metadata to retrieve it. Using metadata to tag a resources allows users to search at a more refined level, and hence more accurately.

There are three principal reasons for using a metadata system:

- Sufficiency: Can a resource be adequately described by the resource itself? For example, an image may contain a picture of a particular geologic structure, but it would be hard to search for this. Words are needed. Although some resources may contain text, they need further information to describe or use them. Not all materials contain inherently adequate self-descriptions.
- Scalability: It is possible to do full text analysis on a single repository with thousands of resources, but it is impractical for large multiple repositories with resources. Metadata provide a highly targeted, rapid search and recovery option at a low cost and greater flexibility.
- Interoperability: The ability for different systems to interchange information, processes and resources is called "interoperability." If different systems can agree to create a mapping between their metadata, then it is possible for each to search one another's metadata. It is also possible for systems to accomplish wide-area searches among many systems if they all have created common mappings. Metadata, as a descriptive system, should allow descriptive mappings among systems—hence, interoperability. Interoperability is important for systems that expect to access resources from a variety of sources [22].

Metadata stored in a system require a schema to structure them. A schema describes what one or more XML documents can look like, and it defines:

- The elements the document contains, and the order in which they appear
- The element content and element attributes, if any

The purpose of schemas is to allow machine validation of document structure. Instead of using the syntax of XML 1.0 DTD declarations, schema definitions use XML element syntax. A correct XML schema definition, therefore, is a well-formed XML document [46].

Research commenced with the study of the work carried out by the organizational bodies associated with generating standards for metadata. The IEEE, ADL, IMS, and AICC all contribute to the LOM standard and Dublin Core (DC) work with the Dublin Core Metadata element set. The extension of the LOM v1.0 metadata schema is covered stating the reason for the necessary extensions to appropriately accommodate the tagging requirements of the metadata repository. The problems that arose from extending the LOM v1.0, namely an ambiguous ontology and methodology, resulted in the base schema being discarded and a new subset schema being drafted.

8.7.1 Dublin Core Metadata Initiative

The Dublin Core Metadata Initiative (DCMI) is an open forum engaged in the development of interoperable on-line metadata standards that support a broad range of purposes and business models. The DCMIs activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices. The Dublin Core Metadata Element Set (DCMES) was the first metadata standard developed out of the DCMI as an IETF—Internet Engineering Task Force—standard. The DCMES provides a semantic vocabulary for describing "core" information properties, such as "Description" and "Creator" and "Date" [47].

Dublin Core metadata are used to supplement existing methods for searching and indexing Web-based metadata. Most DCMI participants are involved in large-scale archiving or cataloging projects that require the use of Dublin Core metadata to enable large collections of object "resources" to be grouped, named, classified, and indexed in a useful fashion.

There are 15 elements in the DC metadata set, and each of these elements has 10 attributes associated with it. Of all of the groups that are creating standards for metadata, this is the largest number of attributes associated with any one set.

The Dublin Core metadata set was the original metadata set from which all other metadata sets stemmed. Groups like the IEEE, AICC, IMS, ADL, ARIADNE, ALIC, and many more based their metadata sets on work carried out by the Dublin Core. At the time of the research, work conducted by the aforementioned groups has progressed significantly, and metadata standards from each of these individual groups were developed or at some stage of development. These metadata sets were better equipped to handle the fast evolution of the standard creation process. From a working point of view, no advantage was apparent from taking the Dublin Core metadata set and implementing it within the scope of the SME Learning Management System.

The Dublin Core metadata set did not sufficiently describe a learning object within the scope of the SME repository. Options for extending the metadata set were not apparent, and altering the set would make it un-interoperable. Another metadata standard was required, leading to the IEEE LOM.

8.7.2 Modifying the IEEE Learning Object Metadata (LOM)

The LOM standard is meant to provide a semantic model for describing properties of the learning objects themselves, rather than detailing ways in which these learning objects may be used to support learning. The LOM indicates the legal values and informal semantics of the metadata elements, their dependencies on each other, and how they are assembled into a larger structure. LOM has specifically been designed to be extendable to accommodate future growth or individual adaptation. The LOM information structures are support metadata exchange, and are neither specifications of an implementation nor specifications of a user interface. The

LOM does not define recommendations concerning bindings or implementations of metadata in representations or notations.

The LOM data model is a hierarchy of data elements, including aggregate data elements and simple data elements (leaf nodes of the hierarchy). In the LOMv1.0 base schema [3], only leaf nodes have individual values defined through their associated value space and data type. Aggregates in the LOMv1.0 base schema do not have individual values. Consequently, they have no value space or data type.

An outline of the LOM metadata mapping is shown in Figure 8.3. The LOM structure is composed of nine elements, which in turn break down into a series of subelements, making up the complete model. Initially in our implementation we

1 General
- 1.1 Identifier
  - 1.1.1 Catalog
  - 1.1.2 Entry
- 1.2 Title
- 1.3 Language
- 1.4 Description
- 1.5 Keyword
- 1.6 Coverage
- 1.7 Structure
- 1.8 Aggregation Level

2 Life Cycle
- 2.1 Version
- 2.2 Status
- 2.3 Contribute
  - 2.3.1 Role
  - 2.3.2 Entity
  - 2.3.3 Date

3 Meta-Metadata
- 3.1 Identifier
  - 3.1.1 Catalog
  - 3.1.2 Entry
- 3.2 Contribute
  - 3.2.1 Role
  - 3.2.2 Entity
  - 3.2.3 Date
- 3.3 Metadata Schema
- 3.4 Language

4 Technical
- 4.1 Format
- 4.2 Size
- 4.3 Location
- 4.4 Requirements
  - 4.4.1 OrComposite
    - 4.4.1.1 Type
    - 4.4.1.2 Name
    - 4.4.1.3 Minimum Version
    - 4.4.1.4 Maximum Version
- 4.5 Installation Remarks
- 4.6 Other Platform Requirements
- 4.7 Duration

5 Educational
- 5.1 Interactivity Type
- 5.2 Learning Resource Type
- 5.3 Interactivity Level
- 5.4 Semantic Density
- 5.5 Intended and user role
- 5.6 Context
- 5.7 Typical Age Range
- 5.8 Difficulty
- 5.9 Typical Learning Time
- 5.10 Description
- 5.11 Language

6 Rights
- 6.1 Cost
- 6.2 Copyright and Other Restrictions
- 6.3 Description

7 Relation
- 7.1 Kind
- 7.2 Resource
  - 7.2.1 Identifier
    - 7.2.1.1 Catalog
    - 7.2.1.2 Entry
  - 7.2.2 Description

8 Annotation
- 8.1 Entity
- 8.2 Date
- 8.3 Description

9 Classification
- 9.1 Purpose
- 9.2 TexonPath
  - 9.2.1 Source
  - 9.2.2 Taxon
    - 9.2.2.1 Id
    - 9.2.2.2 Entry
- 9.3 Description
- 9.4 Keyword

FIGURE 8.3. LOM version 1.0 overview model.

proposed extending the LOM to meet the special needs of the SME environment, building from the base scheme defined in the released version of the standard IEEE 1484.12.1 in July 2002.

Our metadata were designed to be an application profile of the LOM standard. Some extensions were made where LOM was insufficient for the specific purposes of an SME-based repository. The original LOM metadata elements were not replaced or changed; they were taken as they were defined in the standard. Not all of the LOM metadata elements had significance for the goals of SME training, and so some were not used. Those that were used were not changed to maintain conformance to the standard. According to LOM, there can be extension elements, but none of the LOM elements or subelements can be replaced or transformed in any way, so the LOM metadata allow for extensions, but only if the original LOM elements are retained as they were originally defined. The only exception is the possibility to use other values in the Value space than the values defined in the Vocabulary of the Data type of that metadata element.

Originally it was felt that the LOM model did not provide a sufficient level of granularity in identifying learning objects within an SME context, so an extension of the LOM was pursued. There are essentially three ways of extending the metadata schema to suit the particular needs of the system:

1. Creating extensions to the metadata schema that do not overwrite the original schema.
2. Modifying or changing the vocabulary used in the LOM elements.
3. Using classification systems in Category 9 Classification.

Our initial base scheme for the SMEs proposed a number of such extensions, driven primarily by the requirement of increased granularity, and also a specific domain orientation toward SME education and training. Category 1, Category 4, and Category 5 saw the most significant changes based on the original outline (Fig. 8.4 highlighted entries).

Changes in Category 1, General, were primarily focused on more precise definition of the area of application of the learning object. Additional elements, modeled from ARIADNE metadata version 3.0 [48], such as 1.9: Discipline; 1.10: Subdiscipline; 1.11: MainConcept; 1.12: MainConceptSyn; and 1.13: OtherConcepts, were added. Changes in Category 4, Technical, referred to providing a better technical definition of the requirements of the learning object, with extensions in 4.4: Requirement, and several subelements of 4.4, and with 4.8: Material Description being added. The most significant change was in Category 5, Education, with the proposed addition of 5.12: TrainingActivity; 5.12.1: DeliveryMethod; 5.12.2: Time dependence; 5.12.3: Loc dependence; 5.13: Evaluation; 5.13.1: Assessment; 5.13.2: Method; 5.13.3: Number; 5.14: Registration; 5.15: Pre-requisite; 5.16: Qualification; 5.17: Pedagogy; and 5.18: Course-Level—in order to better classify the educational or pedagogic characteristics of the learning object. Much of the change in Category 5, Education, was modeled on proposed changes to metadata schema by both the CUBER [49] and GEMSTONES [50] metadata projects.

1 General
- 1.1 Identifier
  - 1.1.1 Catalog
  - 1.1.2 Entry
- 1.2 Title
- 1.3 Language
- 1.4 Description
- 1.5 Keyword
- 1.6 Coverage
- 1.7 Structure
- 1.8 Aggregation Level
- 1.9 Discipline
- 1.10 SubDiscipline
- 1.11 MainConcept
- 1.12 MainConceptSyn
- 1.13 OtherConcepts

2 LifeCycle
- 2.1 Version
- 2.2 Status
- 2.3 Contribute
  - 2.3.1 Role
  - 2.3.2 Entity
  - 2.3.3 Date
- 2.4 ValidPeriod
  - 2.4.1 Begin
  - 2.4.2 End
  - 2.4.3 Action

3 Meta-Metadata
- 3.1 Identifier
  - 3.1.1 Catalog
  - 3.1.2 Entry
- 3.2 Contribute
  - 3.2.1 Role
  - 3.2.2 Entity
  - 3.2.3 Date
- 3.3 Metadata Schema
- 3.4 Language

4 Technical
- 4.1 Format
- 4.2 Size
- 4.3 Location
- 4.4 Requirements
  - 4.4.1 OrComposite
    - 4.4.1.1 Type
    - 4.4.1.2 Name
    - 4.4.1.3 Minimum Version
    - 4.4.1.4 Maximum Version
- 4.5 Installation Remarks
- 4.6 Other Platform Requirements
- 4.7 Duration
- 4.8 Material_Description

5 Educational
- 5.1 Interactivity Type
- 5.2 Learning Resource Type
- 5.3 Interactivity Level
- 5.4 Semantic Density
- 5.5 Intended and user role
- 5.6 Context
- 5.7 Typical Age Range
- 5.8 Difficulty
- 5.9 Typical Learning Time
- 5.10 Description
- 5.11 Language
- 5.12 TrainingActivity
  - 5.12.1 DeliveryMethod
  - 5.12.2 Time_dependance
  - 5.12.3 Loc_dependance
- 5.13 Evaluation
- 5.14 Registration
- 5.15 Prerequisites
- 5.16 Qualification
- 5.17 Pedagogy

6 Rights
- 6.1 Cost
- 6.2 Copyright and Other Restrictions
- 6.3 Description

7 Relation
- 7.1 Kind
- 7.2 Resource
  - 7.2.1 Identifier
    - 7.2.1.1 Catalog
    - 7.2.1.2 Entry
  - 7.2.2 Description
- 7.3 Concatenation
  - 7.3.1 Position
  - 7.3.2 Associations

8 Annotation
- 8.1 Entity
- 8.2 Date
- 8.3 Description

9 Classification
- 9.1 Purpose
- 9.2 TexonPath
  - 9.2.1 Source
  - 9.2.2 Taxon
    - 9.2.2.1 Id
    - 9.2.2.2 Entry
- 9.3 Description
- 9.4 Keyword

FIGURE 8.4. Extended LOM metadata overview model.

Next there were also a number of proposed variations in classification, or ontology, from that described in the LOM draft standard. To accommodate increased granularity six aggregation levels were defined, as compared to four in LOM version 1.0. The proposed levels were level 0, Fragment; level 1, Topic; level 2, Lesson; level 3, Module; level 4, Course; and level 5, Curriculum. The aggregation levels were used to describe the differences between study elements within an SME learning environment. There were a number of further proposed changes in ontology and vocabulary from that in LOM v1.0, in order that the semantics of the SME environment more accurately reflect the delivery objectives of the SME learning management program.

8.7.3 Taxonomy Models and Ontology

The LOM did not offer an adequate level of metadata coverage for the population of the SME repository. The LOM was lacking in its definition of aggregation levels, or granularity. The associated level of the LOM did not sufficiently define a SME learning object. The educational requirements of the learning objects were not met. There was a need for finer detail in relation to the training activity of the learning object, as well as the evaluation and prerequisites associated with any given learning object. The overall general information related to the learning object was unclear with regard to discipline and concepts tied to a learning object. The solution was to extend the LOM to meet the needs of the SMEs. All the metadata categories, metadata data elements, and subelements adopted from LOM were used as such; they were not changed because of the notes of conformance in LOM.

The IEEE LOM definition of a learning object allows for an extremely wide variety of granularities. This means that a learning object could be a picture of the Mona Lisa, a document on the Mona Lisa (that includes the picture), a course module on da Vinci, a complete course on art history, or even a 4-year master curriculum on Western culture.

In one sense, this is appropriate, as there are a number of common themes to content learning objects of all sizes. In another sense, though, this vagueness is problematic, as it is clear that authoring, deploying and repurposing are affected by the granularity of the learning object.

To address this problem, a learning object taxonomy was developed to identify the different kinds of learning objects and their constituent parts (Fig. 8.5):

- Fragments are the smallest level in this model. These elements reside at a pure data level. Examples include a single sentence or paragraph, illustration, animation, etc.
- Topics are the next level of granularity. This refers to a single learning objective and constitutes 10 to 15 minutes of learning. Fragments are grouped together to form topics.
- Lessons are next in the taxonomy. Lessons consist of topics grouped together with additional tests or assessments areas included, as well as objectives, overviews, summary, prerequisites, etc. [51]. This other content is not seen as reusable in the

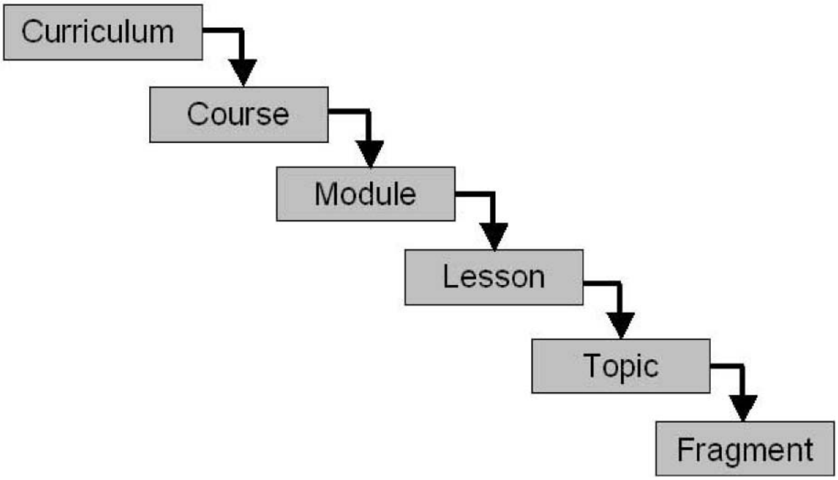


FIGURE 8.5. SME taxonomy model.

strictest sense as it was viewed as being focused on a specific thread or learning concept.
- Modules are a set of lessons that were focused on one study or subject area.
- Courses consist of coherent modules aggregated together.
- Curriculum was a number of courses to provide competence at a designated level in an occupation or profession.

Topics contain fragments. Lessons contain topics. Modules consist of lessons, and courses are made up of separate modules. Curriculum contains courses. The smaller level of granularity in this taxonomy is essential, as research showed that repurposing can only be accommodated by explicitly identifying the information objects and the fragments they contain [13].

CISCO [52] and IMS [17] used tried-and-tested ideas from Open and Distance Learning [53] to determine how to package or collect learning objects together. The smallest unit is a reusable information object (RIO) [54]. It develops a single objective only. CISCO defines each RIO as either being a concept, fact, process, principle, or procedure [52]. Content items and practices (learning activities) are presented to the learner to support that objective. The smallest stand-alone unit is a reusable learning object (RLO), a collection of seven plus-or-minus two RIOs grouped together to teach a common job/task based on a single learning objective. To make the collection of RIOs into a complete learning experience or “lesson,” an overview, summary, and assessment are added to the package. The overview is used to introduce the RLO and act as an advanced organizer for the learner by listing the objective, outline, and job-based scenario for this “lesson.” The summary is used to conclude the RLO and tie the scenario and objectives covered in each RIO together. It also offers a suggested course of action for learners to broaden their knowledge and skills in this area. Finally, the summary is a transition between the RIOs and the final assessment. This structure is drawn directly from ODL.

Comparing the CISCO concept with our SME model in Figure 8.5, the aggregation-level "topic" is equivalent to RIO, and "lesson" is equivalent to the RLO. Any of the higher levels of learning content are seen as a combination of lower material. Additional fields were required to store the summary, outline, aims, objectives, prerequisites, and other fields relating to courses and curricula, but this information was not seen as reusable in the strictest sense as it documented a focus on a specific area only.

A number of issues needed to be better understood if large-scale LO (re)use was to become a reality isues such as aggregation and the notion of design for reuse.

Traditionally, authoring tools mainly support the process of authoring from three points of departure:

- A blank document that needs to be "filled" with content, where the structure of the LO is defined during the elaboration of that content;
- A template that needs to be instantiated, where the structure of the LO is defined a priority;
- An existing LO that is edited and modified in the process of authoring, and then typically saved as a new LO.

The main idea, however, was that learning objects were created by selecting fragments from a repository, usually with the significant assistance of metadata and profiles to do so. These learning objects were then assembled into a new learning object. This was referred to as authoring by aggregation [13].

This new learning object, as it provided new context for the learning, may need to provide "glue" that takes the learner from one learning object to another. A simple example of this kind of facility is the way that presentation authoring tools (like Microsoft Powerpoint, SliTeX, etc.) allow for existing slides to be included in new presentations and then add automatically "next" and "previous" transitions between those slides. More sophisticated "glue" would enable the author of the aggregated learning objects to include transitional material (for example, "In this section, the content will show the concept of inertia that was introduced in Chapter X"), so as to give guidance to the learner on how the components fit together in the aggregate. This kind of "glue" is dealt with by "sequencing" specifications that enable the definition of learning paths. These learning paths are themselves discrete learning objects and as such can be stored separately, modified independenly of the content, reused, AND of course also have their own associated metadata to aid with discovery, search, and retrieval.

Some issues that needed to be taken into consideration when "designing for reuse":

- Ease of modification: The fragments used often depended on the context, and they should be consistent within a given context. The content should be designed in such a way that it becomes easy to alter the information in one fragment, thus producing a new fragment, accessible to all.

- Easily replaced labels: A related issue is that of textual labels in visual material; it should be simple to replace such labels with alternatives, for instance in a different language, or using an alternative vocabulary.
- Adaptive look and feel: Methods need to be developed for adapting the look and feel of content. When different learning objects are aggregated together, the result should not look like a collection of learning objects from different origins. One could think of aggregation tools that allow the author to apply a "design template" to impose a specific look and feel on the resulting aggregate.
- Fragment integration: Fragments within the current repository need to integrate with other fragments with little trouble. This integration can be viewed in the form of a sequential listing of fragments, or within a higher level of granularity. A sequential listing produces a sequence of fragments that form a detailed piece of learning. The higher level of granularity, at a topic level, requires fragments to integrate together to form a more substantial piece of learning content.

It was necessary to have a greater level of granularity that that specified in the LOM. An increase in the level of granularity increased the chance of reusability of learning objects, or pieces of learning objects, and also permitted easier structuring of the learning content. The reusable learning material is below the LO level, and these fragments have little to no context, no formatting, and no specific style. Style and context are added to a learning object via combinations of the design, the learning paths, and/or the presentation layers with typical style sheets [13].

In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them [55]—has been moving from the realm of artificial intelligence laboratories to the desktops of domain experts. An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them [56].

An ontology development process consists of seven steps [57]:

1. Specification: What is the goal of the ontology? What is relevant to fulfill the goal? What needs to be modeled, and what types of granularity are useful?
2. Knowledge acquisition: Collect the information based on the available documents in different data sources. Put this information into a hierarchy structure with respect to the ontology scope. This step occurs in parallel with the specification step.
3. Conceptualization: Concepts in the ontology should be close to objects (physical or logical) and relationships in the related domain. Try to get definition for your ontology from other ontologies.
4. Integration: Integrate the ontology with another ontology if applicable.
5. Implementation: Define the ontology components through an ontology definition language in two stages:
   - Informal stage: sketch the ontology using either natural language descriptions or some diagram techniques.
   - Formal stage: ontology is encoded in a formal knowledge representation language, that is machine computable.

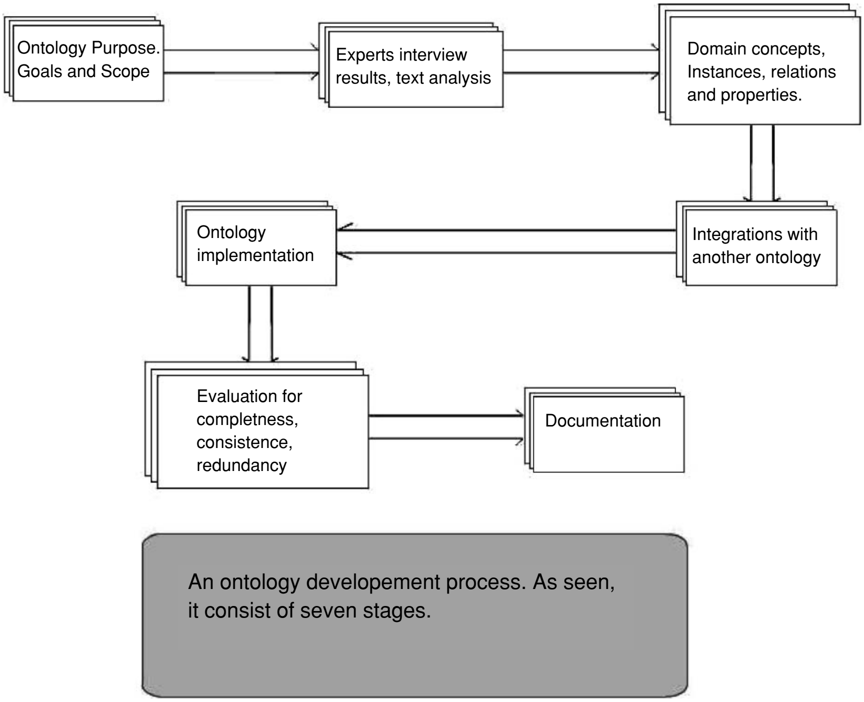


FIGURE 8.6. Ontology development process.

6. Evaluation: Consists of checking for completeness and consistence and avoiding redundancy.
7. Documentation: Produce clear informal and formal documentation. Make the ontology understandable by everyone. An ontology that cannot be understood will not be used.

A number of ontology development tools currently exist; notable among these are Protégé [58], Oiled [59], OntoEdit [60], OntoLingua [61], and WebODE [62].

Most of the tools provide an integrated environment to build and edit ontologies, check for errors and inconsistencies, browse multiple ontologies, and share and reuse existing data by establishing mappings among different ontological entities. However, these tools are influenced by traditional knowledge repository–based ontology engineering methodologies, with steep learning curves, making it cumbersome to use for casual Web ontology development.

The process of ontology development was seen to be an iterative one. Any data elements or subdata elements that were altered or introduced to extend the existing LOMv1.0 metadata schema required an ontology. The ontology defined the terms associated with each data element in the schema. Following from this, cataloging of learning objects within the repository was easier and automated in places. Beyond that, the requirement for greater granularity, and the definition of learning objects in an SME context required that a revised ontology be implemented for existing LOM elements.

Therefore, a new or extended ontology was required for the additional elements added to the schema, as shown in Figure 8.4. Problems arose within the ontology definitions. Redefining the ontology to cater to the additional new elements and expanding existing data element ontology proved difficult. Achieving an unambiguous ontology was not feasible, and therefore automatic metadata generation was impossible. The fast-paced evolution of the e-learning standards made extension of the LOMv1.0 difficult, as changes in the draft version meant upgrading the SME schema to the latest release and starting again. It was decided to work from the opposite end of the scale and approach this problem with a cut-down version of the LOM as opposed to an extended version of the LOM.

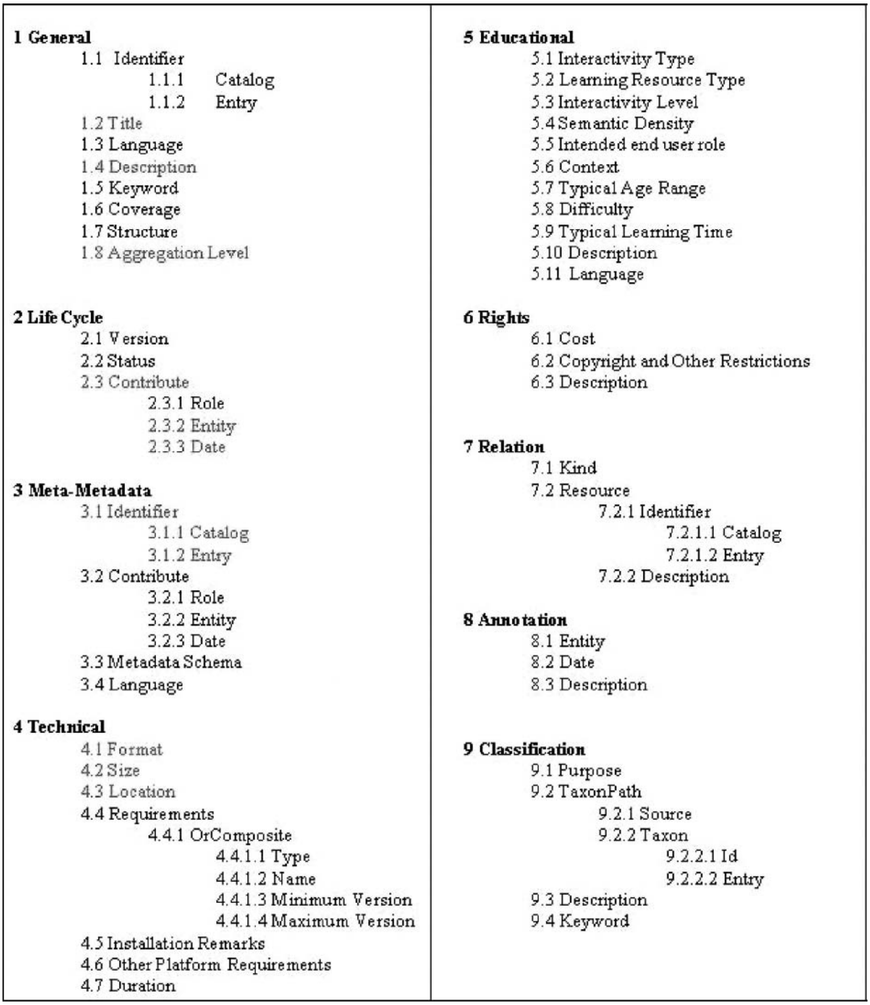
8.7.4 Final Schema of Our System

The initial base schema resulted in additional fields being added to the metadata categories. The drawback of this was that the user, upon uploading content to the repository, was required to insert a lot of information about the learning object. We strongly believed that the users should not have to do this—it should be an automated process or as automated as is humanly possible [63]. Being unable to produce an automated form-filling process for the schema was one of its failings. An ambiguous ontology restricted this automation and resulted in the rejection of this schema as a final version.

An alternative schema, based on the LOMv1.0, was required. The goal was to have the minimal number of fields necessary to adequately describe all learning objects within the central repository. The ADL net [40] listed tables with a variation of the LOM metadata content, and SCORM listed their categories and elements and weighted them with regard to the different levels of granularity. SCORM uses three levels of granularity to define its learning objects: assets, sharable content objects (SCOs), and content aggregation models. Assets within the SCORM represented fragments from a SMEs point of view. SCOs mapped to topics within the levels of aggregation, and everything else was seen as a form of content aggregation.

Initially separate schemas were drafted to accommodate the metadata associated with fragment level and topic level content, as draft no-naming conventions were associated with them. Then an aggregation level data element was inserted into the schema, allowing for the combination of schema to result in a final schema version for the SME metadata. This allowed for the storage of all metadata under the one schema. The elements chosen from the original LOM are shown in Figure 8.7.

The final schema was still standards compliant, and other management systems could access the repositories and search for content based on the metadata information. Figure 8.7 shows the final schema in relation to the LOM v1.0. The required data elements are highlighted in red. This solution permitted an 80 percent automated-tagging process, thus alleviating authors from the necessity of entering known information into the meta-tagging form. The metadata schema allowed for



1 General
- 1.1 Identifier
  - 1.1.1 Catalog
  - 1.1.2 Entry
- 1.2 Title
- 1.3 Language
- 1.4 Description
- 1.5 Keyword
- 1.6 Coverage
- 1.7 Structure
- 1.8 Aggregation Level

2 Life Cycle
- 2.1 Version
- 2.2 Status
- 2.3 Contribute
  - 2.3.1 Role
  - 2.3.2 Entity
  - 2.3.3 Date

3 Meta-Metadata
- 3.1 Identifier
  - 3.1.1 Catalog
  - 3.1.2 Entry
- 3.2 Contribute
  - 3.2.1 Role
  - 3.2.2 Entity
  - 3.2.3 Date
- 3.3 Metadata Schema
- 3.4 Language

4 Technical
- 4.1 Format
- 4.2 Size
- 4.3 Location
- 4.4 Requirements
  - 4.4.1 OrComposite
    - 4.4.1.1 Type
    - 4.4.1.2 Name
    - 4.4.1.3 Minimum Version
    - 4.4.1.4 Maximum Version
- 4.5 Installation Remarks
- 4.6 Other Platform Requirements
- 4.7 Duration

5 Educational
- 5.1 Interactivity Type
- 5.2 Learning Resource Type
- 5.3 Interactivity Level
- 5.4 Semantic Density
- 5.5 Intended end user role
- 5.6 Context
- 5.7 Typical Age Range
- 5.8 Difficulty
- 5.9 Typical Learning Time
- 5.10 Description
- 5.11 Language

6 Rights
- 6.1 Cost
- 6.2 Copyright and Other Restrictions
- 6.3 Description

7 Relation
- 7.1 Kind
- 7.2 Resource
  - 7.2.1 Identifier
    - 7.2.1.1 Catalog
    - 7.2.1.2 Entry
  - 7.2.2 Description

8 Annotation
- 8.1 Entity
- 8.2 Date
- 8.3 Description

9 Classification
- 9.1 Purpose
- 9.2 TaxonPath
  - 9.2.1 Source
  - 9.2.2 Taxon
    - 9.2.2.1 Id
    - 9.2.2.2 Entry
- 9.3 Description
- 9.4 Keyword

FIGURE 8.7. Final schema version for SME based on LOM v1.0.

the population of the SME database and central repository with learning content and objects.

8.8 The Phoenix System

The system that was designed built and tested was called Phoenix, an approximate acronym for PHp Enabled Environment Integrated with XML. It comprised a set of intuitive graphic user interfaces that permitted nontechnical experts to convert electronic content into learning objects and sequence these learning objects into an instructionally sound piece of learning. The objectives of the system were:

- To provide the standard features of any learning content management system
- To facilitate the decomposition of electronic material into smaller pieces of learning termed learning objects
- To dynamically display this content to the end user upon request
- To sequence the learning objects into instructionally sound learning content.

The standard features of any learning management systems exist in Phoenix and aid in the overall functionality of the tool [64,65]. These standard features are necessary for Phoenix to operate properly. For example, in order for the sequence process to work, the search function is required to locate the learning objects that will be utilized in the sequencing procedure.

Standard features include:

1. Access-related features like login and logout, and registration for a new course.
2. Administrative features like modifying user details on the system and maintaining databases.
3. Taking a course and continuing an existing course are basic student requirements.
4. Search capabilities were necessary for the learner, as well as the author, to perform well.

In more detail:

- Users: A definition of the users of the system was required in order to determine what functionality was needed to support them. The users were the learner, the author, and the administrator.
- Login: The users on the system were required to login before accessing any of the material on the Web site. Once users logged in successfully, the functionality of the system was available to them, depending on their access level.
- Logout: If the user closed the browser window, the session was automatically closed and the user was logged out. The session could time-out from inactivity, and the user was prompted to log in again. This was added as a security feature of the tool.
- Search: This function was divided into two sections. The first section was a browse scenario. Authors could browse the content in the repository and select individual fragments. The other option was to search for a specific piece of content. This functionality was to support authors in the finding of learning objects.
- Registration: Learners on the system could register for a new course and, once accepted, could commence taking the course.
- Take a course: After learners logged in, they were presented with the option to continue with an existing course that was partially completed or register for a new course.

- Maintain databases: The administrator on the system maintained the databases and provided support for the authors and learners using the system.

8.8.1 Implementing Phoenix

The best implementation option for possible future extensions or modification was an open-source one. To create a system that was adaptable and extendable, the source code needed to be available so that source code editing could include new system features. Open-source software (OSS), also offered significant cost, reliability, and support advantages that are attractive to SMEs. The Phoenix system consists of several layers that work together to form the overall system. These layers include the base layer, the search layer, the dynamic delivery layer, and the management layer. A base layer requires the following features:

- An operating system capable of running the server and handling at least a database server, a Web server, and a mail server.
- A stable Web server that was capable of handling multiple requests for numerous users.
- A database server to handle the metadata, and store the information pertaining to the users accessing the system. Content stored locally on dynamically created folders based on the users accounts.
- A scripting language that was capable of interacting easily with the Web server as well as the database server and operating system. It should be robust and have fast access times based on execution of code and be easily portable to other systems (for backing up systems or mirroring sites to disperse the server load).

Our final system was based on the established LAMP technology: Linux (operating system), Apache (Web server), MySQL (database server) and PHP (scripting language) [66].

The search layer of Phoenix divides into a browsing process and a searching process. The browse permits an author to browse through the content in the repository; published files in the repository are displayed for the author to see. Anything suitable can be selected and aggregated by the LO being authored.

Alternatively, authors may search the metadata for a specific learning object under headings determined by the metadata schema design and select relevant pieces of learning for use within a learning object. This search accesses all levels of granularity.

The administrative layer, accessed through a Web interface, allows administrators to perform two main tasks: authorize new users and change the user access levels. Administrators may also activate and deactivate accounts.

Dynamic delivery is an important feature of any learning management system. E-learning is designed with just-in-time or just-enough learning in mind. Users take courses at their own speed or access material on-line for a specific answer or piece of information. Metadata and standards influence the dynamic delivery of content. The tagging process and storage of the learning objects in a central repository permit the reusability of the content. Reuse of the learning objects assists in the dynamic delivery process.

8.9 Phoenix System Architecture and Functionality

Phoenix was required to facilitate the creation, storage, and publishing of content by nontechnical users, to include LO sequencing into topics and courses and the dynamic delivery of learning content. The technical layers, that is, the Web server, the database server, the operating system, etc., needed to integrate with current IT environments, without requiring the purchase of additional hardware and software. Finally the system needed to support the administration of all users [67].

Each element of the LAMP acronym provided an essential layer of functionality (Fig. 8.8):

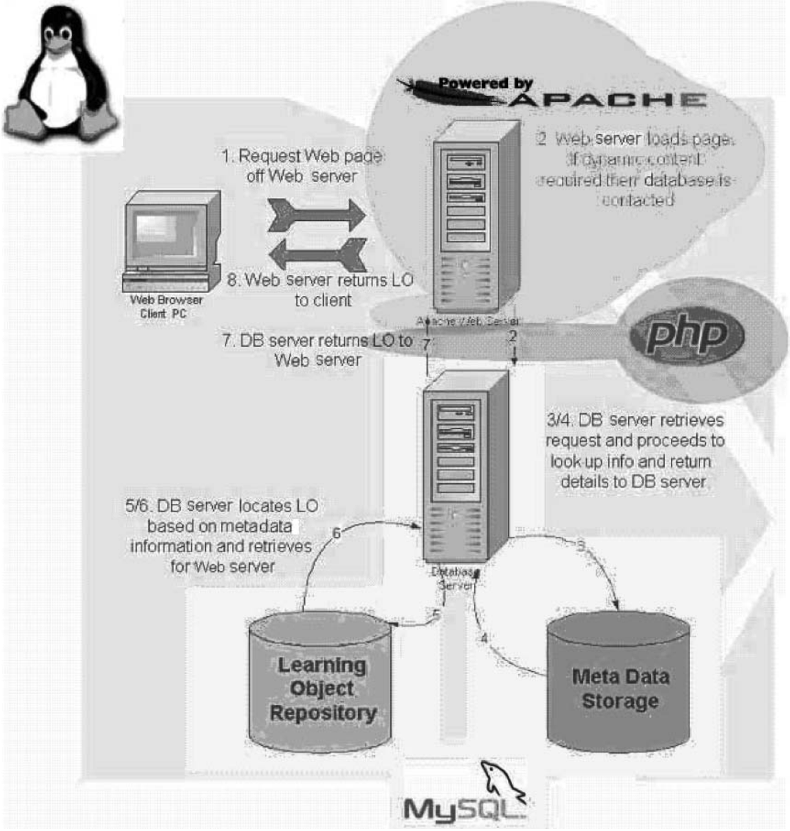


FIGURE 8.8. Phoenix overall structure based on LAMP technology.

- Linux is the operating system. Linux has grown into a reliable operating system that now gets corporate support from start-ups like Red Hat and big companies like IBM.
- Apache is the Web server, the world's most-used Web server. It is controlled by a group called the Apache Software Foundation and has also been embedded in commercial products like IBM WebSphere.
- MySQL is the DBMS (database management system). The MySQL database server is the world's most popular open-source database. With more than five million active installations, MySQL has quickly become the core of many high-volume, business-critical applications. Customers such as Yahoo!, Google, Cisco, Sabre Holdings, HP, and NASA are realizing significant cost savings by using MySQL's high-performance, reliable database management software to power large Web sites, business-critical enterprise applications, and packaged software applications [68];
- PHP is an object-oriented web scripting language. It's similar to Java Server Pages (JSP) and Microsoft Active Server Pages (ASP). PHP is another Web-scripting technology that mixes HyperText Markup Language display code with programming instructions.

8.9.1 Unique Features for the SMEs

Several unique elements were coded and implemented in Phoenix specifically for SMEs. They respond to the recognition that in SMEs and other small-scale users, content authoring and aggregation are likely to be carried out by people who are not trained educators These elements included an upload tool, an authoring section, and an aggregation environment. Learning material can be uploaded as either a fragment or a topic. A fragment, the lowest level of granularity, consists of raw data elements, images, text, movie clips, etc. Fragments are selected from elsewhere on the PC or network and submitted to Phoenix. The system determines if the material is valid: the file size is not zero bytes, the file has an acceptable format, and the file name is not already used or exists already in the database.

If all is OK, the file is stored in the repository and the author is asked to fill in the metadata form (Fig. 8.9). Meeting our requirement for minimum form filling, 80% of the fields are automatically completed. The author only needs to supply the remaining 20%. The author is shown a preview to verify that the correct material is being uploaded.

Assembling fragments into topics is done by building a composite knowledge object (CKO) (Fig. 8.10). Again a user-friendly form-filling format is used. The form permits the insertion of content, text, audio, images, etc., between or around fragments. It also allows existing topics to be edited to form new ones—a very useful reuse feature for authors.

The CKO tool interrogates the metadata database and displays any fragments that the author calls up. The necessary additional metadata fields are part of the form. The CKO creation process uses an open-source what-you-see-is-what-you-get

Fragment Upload

Location: Phoenix > Upload > Upload Frag

Logged in as kjohnson.

Metadata Fields

Title: sunset

Description:

Author: Kevin Johnson

Date: Fri-11-June-2004, 15:44

Format: jpg

Size: 71189 bytes

Location: /var/www/html/content/kjohnson/fragment/sunset.jpg

Copyright Upload: No

Uploaded Fragment

FIGURE 8.9. Upload fragment screenshot.

(WYSIWYG) on-line HTML editor called, solmetra PHP asp.net wysiwyg (SPAW)[1] [69]. Through SPAW authors can edit and reedit a topic until satisfied it is ready for publishing.

This aggregation tool permits the sequencing of fragments to form topics, topics to form courses, and so on. Adding a new topic to the system requires the execution of five steps. The first step creates the topic name, and description, the author, creation date, size, and aggregation level are autogenerated. The author determined the copyright issue. Next, fragments are chosen from the central repository. The author must hold the copyright or the fragments must be copyright free and they must be fragments (not some higher level of aggregation). The author has an option to view fragments in a pop-up window (Fig. 8.11).

Step 3 involves the ordering of the selected fragments, the author chooses the first fragment to be displayed followed by the second fragment and so on. Error checking ensures the sequence's uniqueness.

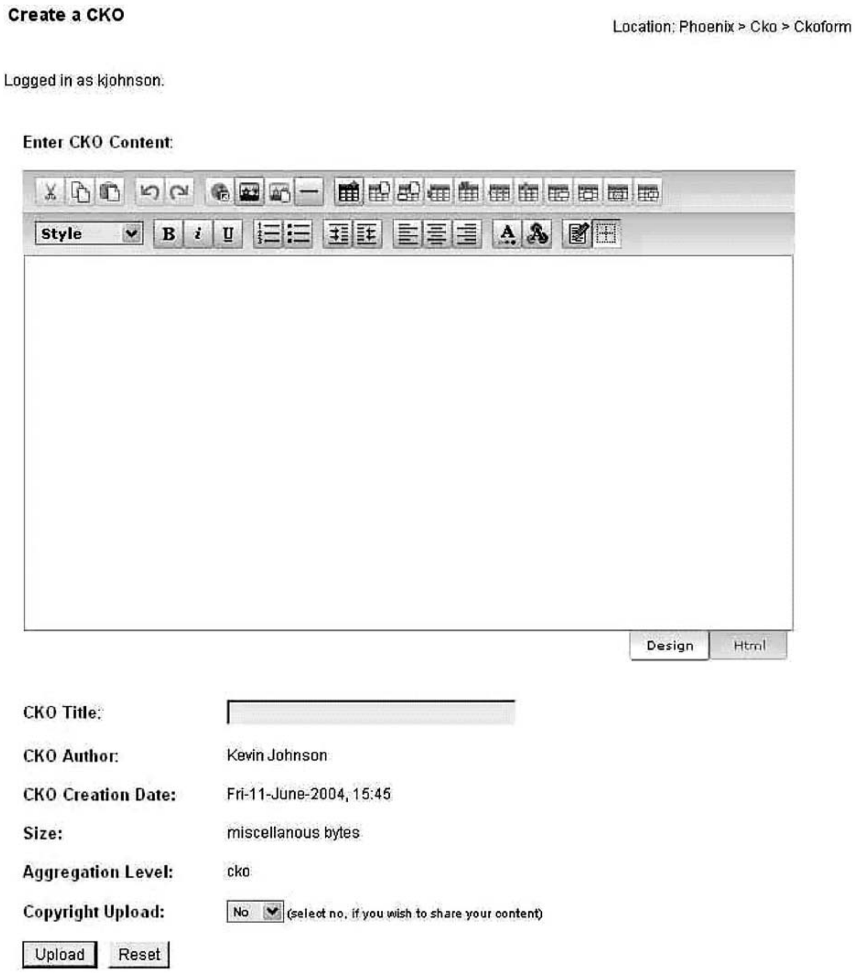
Step 4 verifies step 3 and enables returning to previous steps for re-authoring again. Error checking verifies uniqueness.

Step 5 creates an XML file and its storage in the central repository. The XML holds the ID of all the fragments used within a given topic and the sequencing. This approach enables dynamic delivery to learners. When a learner requests a given topic, the XML is interrogated and the content dynamically gathered and delivered.

The XML file is created in accordance with the IMS simple sequencing specification [38] and is termed a manifest file (Fig. 8.12). A manifest also enables

---

[1] S=solmetra, P=PHP, A=ASP.NET, W=WYSIWYG



Create a CKO

Location: Phoenix > Cko > Ckoform

Logged in as kjohnson.

Enter CKO Content:

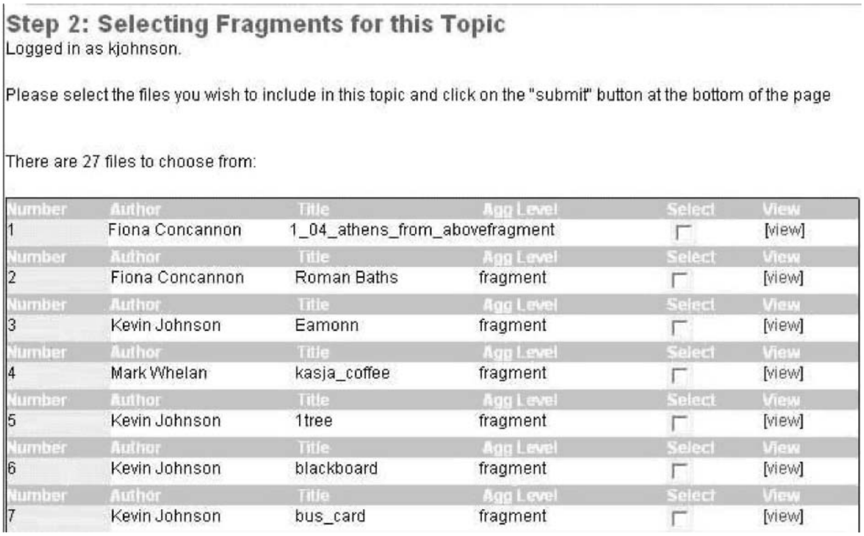
CKO Title:	
CKO Author:	Kevin Johnson
CKO Creation Date:	Fri-11-June-2004, 15:45
Size:	miscellanous bytes
Aggregation Level:	cko
Copyright Upload:	No (select no, if you wish to share your content)

FIGURE 8.10. Composite knowledge object (CKO) editor within Phoenix.

interoperability. If a topic is to be exported, its XML file is scanned and the fragments referenced are collected, packaged, and compressed into a single file ready for transport, in conformance with the IMS content packaging specification [17, 70]. Interoperability of standards permits this process to execute successfully.

8.10 Delivery, Evaluation, and Results

The Phoenix tool was designed to be rapidly adaptable to the needs of any specific learning environment. Its open-source nature enables this. Some might say that the dotLRN [71] system is very similar to the Phoenix system, but it was necessary to create our own system for several reasons. Foremost of these included being able to



Step 2: Selecting Fragments for this Topic
Logged in as kjohnson.

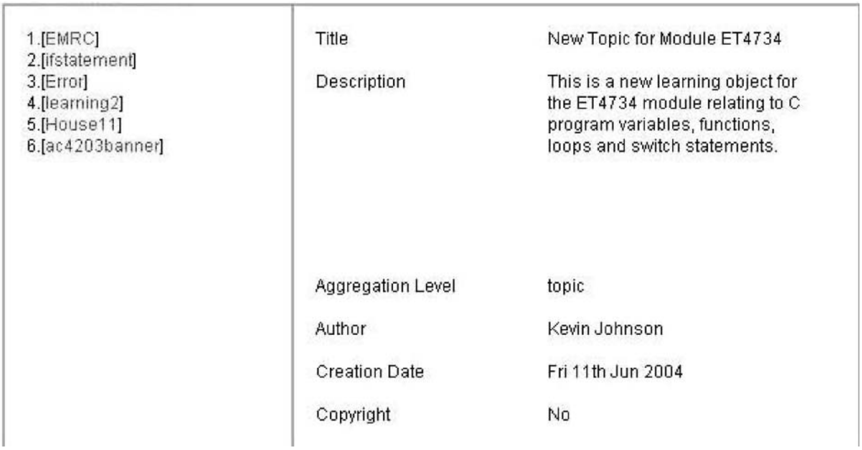
Please select the files you wish to include in this topic and click on the "submit" button at the bottom of the page

There are 27 files to choose from:

Number	Author	Title	Agg Level	Select	View
1	Fiona Concannon	1_04_athens_from_abovefragment		☐	[view]
Number	Author	Title	Agg Level	Select	View
2	Fiona Concannon	Roman Baths	fragment	☐	[view]
Number	Author	Title	Agg Level	Select	View
3	Kevin Johnson	Eamonn	fragment	☐	[view]
Number	Author	Title	Agg Level	Select	View
4	Mark Whelan	kasja_coffee	fragment	☐	[view]
Number	Author	Title	Agg Level	Select	View
5	Kevin Johnson	1tree	fragment	☐	[view]
Number	Author	Title	Agg Level	Select	View
6	Kevin Johnson	blackboard	fragment	☐	[view]
Number	Author	Title	Agg Level	Select	View
7	Kevin Johnson	bus_card	fragment	☐	[view]

FIGURE 8.11. Selection process within the Phoenix environment.

implement the above schema that resulted from the study of the needs of the SMEs. While the dotLRN is built on open-source technology also, the level of understand and technology savvy required to operate and maintain the system is higher than that of a standard LAMP build. The code associated with dotLRN is also harder to manipulate and understand as opposed to PHP and MySQL. Our initial testing of the open-source concept was carried out in an on-campus university environment, as opposed to within an SME, so as to enable better monitoring and control.



1.[EMRC]
2.[ifstatement]
3.[Error]
4.[learning2]
5.[House11]
6.[ac4203banner]

Title	New Topic for Module ET4734
Description	This is a new learning object for the ET4734 module relating to C program variables, functions, loops and switch statements.
Aggregation Level	topic
Author	Kevin Johnson
Creation Date	Fri 11th Jun 2004
Copyright	No

FIGURE 8.12. XML-based output from the Phoenix system.

We chose a course based on the constructivist cognitive apprenticeship model with learners who were new to e-learning. In the cognitive apprenticeship model, parallels are made with the teaching tradition of apprenticeship and schooling. Alan Collins, John Seely Brown, and Ann Holum [73] propose that students learn best when the thinking is made visible. Traditional apprenticeship focuses on the combination of observation, coaching, and scaffolding. Our aim was to imitate this successful form of learning in the more controllable university environment. The design used the apprenticeship model through adaptive learning guides posing as superheroes. The superhero related his/her power or weakness to a creative writing technique and thus serves, as both a guide and a mnemonic device.

Two student groups were phased sequentially. The Phoenix-based system was adapted to meet specific course needs in each of these phases. Adaptations were carried out by code modification of the base system, access to the source code being enabled by the open-source nature of Phoenix.

The student group in Phase One consisted of 17 students studying a course in electronic production over one semester. Their comments and reactions were solicited by email and by on-line questionnaires (Fig. 8.13).

In phase one most students were satisfied with the LMS and the on-line course. They suggested the following:

- Increased file size for uploading assignments
- More sample assignments and links to relevant Web sites
- More comments on corrected assignments
- Email notification to lecturer/TA when assignments have been uploaded
- Better access to information on the assignment titles

The exact nature of these suggestions is not of direct importance, but they illustrate areas for adaptation or improvement in the system to better meet the

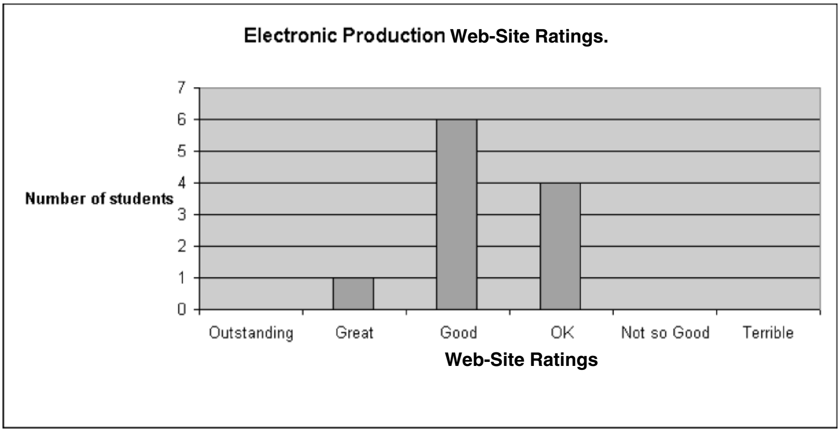


FIGURE 8.13. Phase one feedback information.

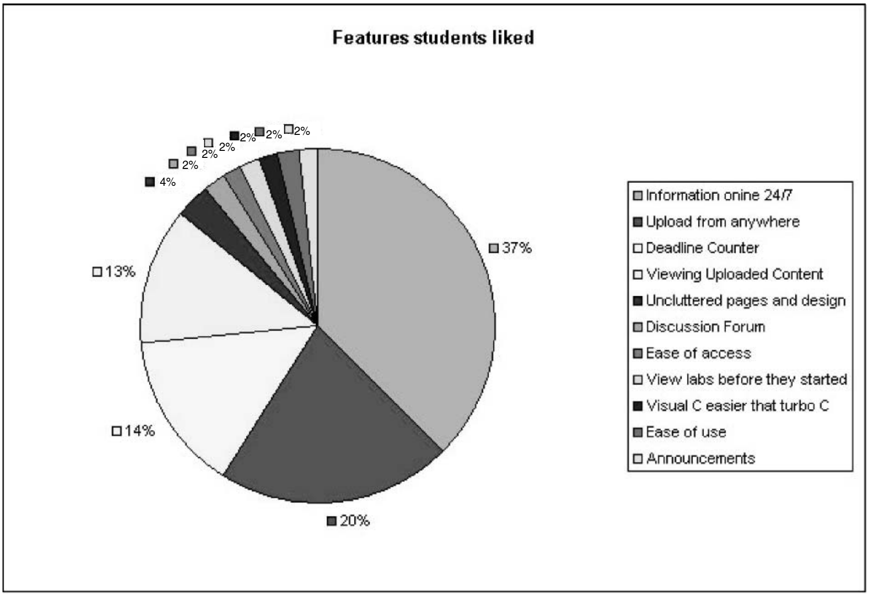


FIGURE 8.14. Phase two features feedback.

needs of the learners. The information gathered provided a means to create further evolution of the code and extend the functionality of the tool to better suit the requirements of the learners and the author [72]. For example, a deadline counter was added to let students know by when assignments must be uploaded.

In phase two, 50 students following a different course, again over a semester, in general approved of this style of learning and the delivery approach. They liked having access to the notes on-line anytime, anywhere, liked that labs could be uploaded from anywhere, liked the new deadline counter feature, and liked the upload viewer (a feature that was added partway through the semester, again illustrating the power of the use of open source code) (Fig. 8.14).

The students in phase two had many more suggestions for additional or improved features (Fig. 8.15).

Although much of the student feedback in both phases related to their impressions of this style of learning and would apply to almost any LMS/LCMS, we found the rapid adaptability afforded by direct access to the code of Phoenix a uniquely advantageous feature. It enabled us to add desired new functionality rapidly and accurately, and indicated that tailoring to the needs of specific SMEs would be practical. We set out to create a system that was adaptable, flexible, extensible, and inexpensive that met the needs of the learners. Our test confirmed we have achieved this. Access to the source code permitted the numerous updates to the system, resulting in improved variants of the Phoenix tool. The open-source choice justified itself. Phoenix proved rapidly adaptable to different learning scenarios and was responsive to the needs of both novice and the more experienced users—both teachers and learners. The basic system was robust and responsive.

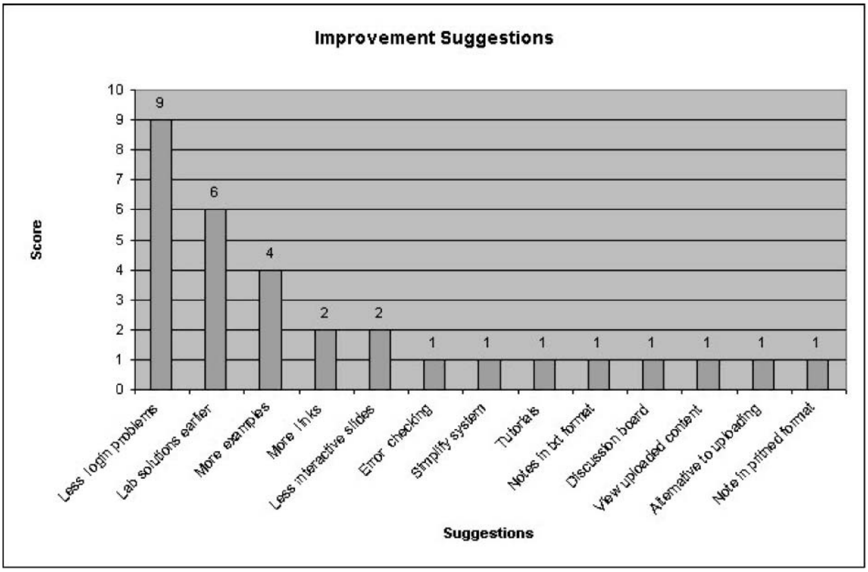


FIGURE 8.15. Phase two improvement suggestions.

8.11 Conclusion

The main task we undertook was to make an LCMS with an embedded e-learning content creation tool for use by nonexpert authors and suitable for use in SME or other small-scale situations. The resulting Phoenix system was based on a novel metadata schema with clearly defined ontology focused on an implementation using reusable and interoperable e-learning learning objects. It supports the creation of instructionally sound e-learning content, and provides the functionality to aggregate and sequence this content into larger learning structures. The tool's attributes included being flexible, adaptable, customisable, standards compliant, using reusable learning objects, and inexpensive.

The main achievements include:

1. A successful standards-based interoperable compliant metadata set permitting the search and retrieval of any learning objects within the repository
2. An operational open source–based implementation successfully tested for functionality and robustness on two separate groups of learners over a one-year period
3. An adaptable learning system open to future modifications. These separate testing stages highlight the extensibility and adaptability of the system to meet the users' requirements and cater to their needs.

Phoenix encompasses the following attributes: flexible, adaptive, customizable, standards compliant, uses re-usable LOs, and low cost, making it ideal for its intended use in small companies or other small-scale users. Additional tools have subsequently been added to include a training or learning needs analysis phase.

References

1. Hokanson, B., Hooper, S. (2000) *Computers as cognitive media: the potential of computers in educations.* Computers in Human Behaviour, 16(5):537–552.
2. SCORM. http://www.adlnet.org/.
3. IEEE. (2002) IEEE LOM Version 1, final draft. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.
4. IMS. (2001) http://www.imsproject.org.
5. WebCT. (1997) WebCT—e-learning solutions for higher education. http://www.webct.com.
6. Blackboard. (1997) Blackboard Software Company. http://www.blackboard.com.
7. TopClass. (1995) TopClass Computer Systems. http://www.wbtsystems.com.
8. Docent Docent Software Company. http://www.docent.com.
9. TrainerSoft TrainerSoft Company. http://www.trainersoft.com.
10. Fallon, C., Brown, S. (2003) *e-Learning Standards: A Guide to Purchasing, Developing and Deploying Standards—Conformant e-Learning*. Boca Raton, Florida: St. Lucie Press.
11. Avaltus. (2000) http://www.avaltus.com/.
12. Brahler, C.J., Peterson, N.S., Johnson, E.C. (1999) *Developing on-line learning materials for higher education: An overview of current issues.* Educational Technology and Society, 2(2):42–54.
13. Duval, E., Hodgins, W. (2003) *A LOM research agenda*. In: *WWW2003*. Budapest, Hungary.
14. Rust, G., Bide, M. (2000) The “indecs” metadata framework: principles, model and data dictionary. http://www.indecs.org/pdf/framework.pdf. Indecs Framework Ltd in 2000.
15. Sciore, E., Siegel, M., Rosenthal, A. (1994) *Using semantic values to facilitate interoperability among heterogeneous information systems.* ACM Transactions on Database Systems (TODS), 19(2):254–290.
16. Drew, P., et al. (1993) *Report of the workshop on semantic heterogeneity and interoperation in multidatabase systems.* ACM Sigmod Record, 22(3):47–56.
17. IEEE. (1999) Institute of Electrical and Electronic Engineers. http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=about/whatis&file=index.xml&xsl=generic.xsl.
18. AICC. (1988) http://www.aicc.org/.
19. ARIADNE. (2001) European ARIADNE Project. http://www.ariadne-eu.org.
20. UCGIS. (1998) University Consortium for Geographic Information Science (UCGIS) Research Priority White Papers—Paper 5: Interoperability of Geographic Information. http://www.ncgia.ucsb.edu/other/ucgis/research_priorities/paper5.html.
21. Wiley, D.A. (2000) *Learning object design and sequencing theory*. In: *Instructional Psychology and Technology Department*. Provo City, UT: Brigham Young University, p. 142.
22. Tannenbaum, A. (1991) *Computer Networks*, 2nd ed. New York: Prentice-Hall.
23. Padrick, N. (2003) *Information you can use: A Data Mart Primer.* Solutions Journal, 9(2):23–27.
24. Dushay, N. (2002) *Localising experience of digital content via structural metadata.* In: *International Conference on Digital Libraries. Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, Portland, Oregan. New York: ACM Press.
25. LTSC. (2002) IEEE *Learning Technology Standards Committee Mission.* http://grouper.ieee.org/groups/ltsc/index.html.

26. Gibbons, A.S., Nelson, J., Richards, R. (2000) *The nature and origin of instructional objects*. In: *The Instructional Use of Learning Objects: Online Version.*, Wiley, D.A., ed., Agency for Instructional Technology. available at http://www.reusability.org/read/chapters/gibbsons.doc
27. Hodgins, W. (2000) Into the future. http://www.learnativity.com/download/MP7.PDF.
28. Dahl, O.J., Nygaard, K. (1966) *SIMULA—an algol based simulation language.* Communications of the ACM, 9(9):671–678.
29. Reigeluth, C.M., Nelson, L.M. (1997) *A new paradigm of ISD?* Educational Media and Technology Yearbook, 22:24–35.
30. LOM. (2002) IEEE LTSC WG12—Learning object metadata. http://ltsc.ieee.org/wg12/.
31. Newsroom. (1999) Internet growing too fast for search engines. http://www.editors-service.com/articlearchive/search99.html.
32. Cisco. (2000) Reusable learning object strategy: definition, creation process, and guidelines for building. http://www.cisco.com/.
33. Hodgins, W. (2000) Everything you ever wanted to know about learning standards but were afraid to ask. http://www.linezine.com/2.1/features/wheyewtkls.htm.
34. Wagner, E.D. (2002) The new frontier of learning object design. http://www.elearningguild.com.
35. Wiley, D.A. (2000) Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy. http://reusability.org/read/chapters/wiley.doc.
36. ISO. (2002) Information technology: learning by IT. http://jtc1sc36.org/doc/36N0264.pdf.
37. Friesen, N., McGreal, R. (2002) International Review of Research in Open and Distance Learning. http://www.irrodl.org/content/v3.2/tech11.html.
38. IMS. (2001) IMS content packaging (CP). http://www.imsproject.org/content/packaging/index.cfm.
39. UNFOLD. (2003) The UNFOLD Project. http://www.unfold-project.net/.
40. ADLNet. (2000) Advanced Distributed Learning Network (ADL Net) *Advanced Distributed Learning, SCORM Past.* http://www.adlnet.org/index.cfm?fuseaction=scormhist.
41. IMS. (2001) IMS simple sequencing (SS). http://www.imsproject.org/simplesequencing/index.cfm.
42. ADL. (2003) Advanced distributed learning network (ADL Net). *ADL Co-Labs: overview.* http://www.adlnet.org/index.cfm?fuseaction=colabovr.
43. Ariadne. (2002) ARIADNE Foundation Presentation 1.1 http://www.ariadne-eu.org/en/about/general/benefits/index.html.
44. ISO. (2004) About ISO, introduction. http://www.iso.org/iso/en/aboutiso/introduction/index.html.
45. IEC/CEI. (2004) International Electrotechnical Commission. http://www.iec.ch/.
46. ISO/IEC. (2004) International Organisation for Standardisation/ International Electrotechnical Commission JTC1 SC36. http://jtc1sc36.org/.
47. ALIC. (2001) Advanced Learning Infrastructure Consortium. http://www.alic.gr.jp/eng/.
48. XMLSPY. (2000) http://www.xmlspy.com/.
49. DublinCore. (1998) Dublin Core metadata element set, version 1.1—reference description. http://dublincore.org/documents/1999/07/02/dces/.
50. Ariadne. (1999) ARIADNE metadata recommendation version 3.0. http://ariadne.unil.ch/Metadata/ariadne_metadata_v3final1.htm.

51. CUBER. (1999) EU 5th Framework IST Programme, Personalised Curriculum Builder in the Federated Virtual University of the Europe of Regions, final version of metadata specification.http://www.cuber.net/web-v1/publications/cuber-d9-1.pdf.
52. GEMSTONES GESTALT, ACTS (Advanced Communications Technologies and Services) project, courseware metadata design V3 (GEMSTONES). http://www.fdgroup.com/gestalt/D0401_3.pdf.
53. Moodle Moodle Web site. http://moodle.org.
54. Gruber, T.R. (1993) *A translation approach to portable ontology specification.* Knowledge Acquisition, **5**:199–220.
55. Noy, N.F., McGuinness, D.L. (1999) Ontology developement 101: a guide to creating your first ontology. http://www.standford.edu/.
56. Poggi, A., Bergenti, F. (2000) *Multi-agent systems: ontology*. In: *ESAW Workshop at ECAI 2000*, London, England.
57. Protege. (2000) Technical report: using Protégé-2000 to edit RDF. http://www.smi.Stanford.edu/projects/protege/protegerdf/protege-rdf.html.
58. Bechhofer, S., et al. (2001) *OilEd: a reasonable ontology editor for the semantic Web.* In: *KI2001, Joint German/Austrian conference on artificial intelligence*, Vienna, Austria.
59. Sure, Y., et al. (2002) *OntoEdit: collaborative ontology development for the semantic Web*. In: *International Semantic Web Conference (ISWC02)*, Sardinia, Italy.
60. Farquhar, A., Fikes, R., Rice, J. (1996) *The Ontolingua server: a tool for collaborative ontology construction*. In: *10th Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.
61. Arpírez, J.C., et al. (2001) *WebODE: a scalable ontological engineering workbench*. In: *First International Conference on Knowledge Capture (K-CAP 2001),* Victoria, Canada.
62. Duval, E., et al. (2002) Metadata principles and practicalities. http://dlib.org/dlib/april02/weibel/04weibel.html.
63. Brennan, M., Funke, S., Anderson, C. (2001) Learning Content Management Systems: a new e-learning market segment emerges. http://www.idc.com.
64. Concannon, F., Johnson. K. (2002) *Learning through learning content management systems*. In: *Human Computer Interaction, HCI*, Brighton, England.
65. Cross, J., Hamilton, I. (2002) The DNA of E-learning. http://www.austrainer.com/elearning/dna-of-elearning.htm.
66. Kearns, D. (2004) Open Source all stars. http://www.nwfusion.com/newsletters/nt/2004/0322nt1.html.
67. O'Droma, M., Ganchev, I., McDonnell, F. (2003) *Architectural and functional desgin and evaluation of e-learning VUIS based on the proposed IEEE LTSA reference model.* The Internet and Higher Education, 6(3):263–276.
68. Cisco. (1999) Cisco Systems—reusable information object strategy. http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf.
69. MySQL. (1995) http://www.mysql.com.
70. SPAW. (2002) http://www.solmetra.com/spaw/.
71. dotLRN. (2001) http://dotlrn.org/.
72. IMS. (2001) IMS Global Consortium, Inc *IMS Background.* http://www.imsproject.org/aboutims.cfm.
73. Collins, A., Brown, J.S., et al. (1991) Cognitive Apprenticeship: Making Thinking Visible. American Educator: The American Federation of Teachers, 15(3):6–11, 38–46.