Learning Objects
By Stephen Downes
University of Alberta

23 May, 2000

Abstract

This essay discusses the topic of learning objects in three parts. First, it identifies a need for learning objects and describes their essential components based on this need. Second, drawing on concepts from recent developments in computer science, it describes learning objects from a theoretical perspective. Finally, it describes learning objects in practice, first as they are created or generated by content authors, and second, as they are displayed or used by students and other client groups.

A. The Need for and Nature of Learning Objects

Some Assumptions and a Premise

Before launching directly into a discussion of learning objects, let's start with some assumptions and a premise.

The assumption is this: that there are thousands of colleges and universities, each of which teaches, say, "Introductory Trigonometry". And each trigonometry course in each of these institutions describes, say, the sine wave function.

Moreover – because the properties of sine wave functions remains constant from institution to institution – we can assume that each institution's description of sine wave functions is more or less the same as each other institution's. What we have, then, are thousands of similar descriptions of sine wave functions.

Now suppose that each of these institutions decided to put its "Introductory Trigonometry" course online. This is no stretch; the Institute for Higher Education Policy estimates that 85 percent of four-year colleges will offer courses online by 2002.[1] So now what we have are thousands of similar descriptions of sine wave functions available online.

Now for the premise: the world does not need thousands of similar descriptions of sine wave functions available online. Rather, what the world needs is one – or maybe a dozen, at most – descriptions of sine wave functions available online.

The reasons are manifest. If some educational content, such as a description of sine wave functions, is available online, then it is available worldwide. So even if only one such piece of educational content were created, it could be accessed by each of the thousands of educational institutions teaching the same material.

Moreover, educational content is not inexpensive to produce. Even a plain web page, authored by a mathematics professor, can cost hundreds of dollars. Include graphics and a little animation and the price is double. Add an interactive exercise and the price is quadrupled.

Suppose that *one* description of the sine wave function is produced. A high quality and fully interactive piece of learning material could be produced for, say, a thousand dollars. If a thousand institutions share this one item, the cost is a dollar per institution. But if *each* of a thousand institutions produces a similar item, then each institution must pay a thousand dollars, or the institutions, collectively, must pay a million dollars. For one lesson. In one course.

The economics are relentless. It makes no financial sense to spend millions of dollars producing multiple versions of similar learning objects when single versions of the same objects could be shared at a much lower cost per institution. There will be sharing, because no institution producing its own materials on its own could compete with institutions sharing learning materials.

Courses? No, Not Courses

If we accept the premise that institutions will share learning materials, then we need to ask, what will they share? The answer that intuitively offers itself is: courses.

Listings of online learning materials, say Telecampus[2] or the Web of Asynchronous Learning Networks[3], list only courses. Good listings, they are divided into subject areas, where each subject page contains a list of similar courses offered by different institutions.

These directories are directed at potential consumers of learning material, that is, students. Students are typically motivated by an interest in a topic[4] and select courses from the list of offerings in that topic. Moreover, students are typically *offered* learning materials in course-sized units, and attempt to complete degree of diploma programs defined as sets of related courses.

Why, then, would institutions not share these courses?

To a certain degree, they already do so. Most colleges and universities define course articulation policies, whereby a course completed at one institution is accepted for credit at another institution. A good example is the Baccalaureate Core Course Equivalency defined by Oregon State University for courses at thirteen regional community colleges.[5]

Course articulations are the result of complex negotiations between teams of academics. Consider, for example, the information contained in the Illinois Mathematics and

Computer Science Articulation Guide.[6] To count as equivalent credit for, say, a trigonometry course, a candidate course must require certain pre-requisites and contain material covering a certain set of topics.

Because of the regional nature of course articulations – it is notable that Oregon State University has made no attempt to articulate courses offered by, say, community colleges in Florida – and because of the detailed topic-by-topic definition or articulation agreements, course sharing between institutions is difficult to define and maintain. It is unlikely that *any* course could be shared by any significant number of institutions in different states or different nations.

We see this disparity reflected on online course listings. Returning to the Telecampus guide we find twenty separate history courses listed.[7] No two of the courses share the same name. And though a number of courses focus on the same region and time period, no two of the courses share the same contents. This is true to more or less a degree across all subjects and across all institutions. Although courses may share elements in common, it is rare to find two courses from two institutions that share the same, and only the same, set of elements.

Thus, courses themselves are *not* suitable candidates for sharing. Yet the dominant form of online educational today is the course. So it should come as no surprise that there is very little sharing of educational resources, even online resources, despite the tremendous cost savings.

Despite what the world needs, what the world is getting is a thousand different versions of "Introductory Trigonometry". It makes no sense, and the current system is going to have to change.

Sharing the Old Way

Whether at the K-12 or college level, today's classroom is already an example of extensive resource sharing. Of course, neither the producers nor the consumers of those resources would describe the transactions as "sharing", nonetheless, if we describe 'sharing' as meaning 'one centrally produced resource used by many', then these classes are sharing resources.

The clearest example of resource sharing "the old way" in today's classrooms occurs through the use of textbooks. These resources bear all the hallmarks of sharing: they are centrally produced and obtained as needed by classroom instructors around the world. In many cases, the information in textbooks is so commonly used the work becomes standard.

But textbooks are just one type of item among many that are shared by classes around the world. No K-12 school is complete without a set of wall maps in geography classes, periodical tables of the elements in science classes, and sets of large block letters for the early years. A rich and useful set of classroom displays is distributed by organizations as varied as astronomical societies[8], museums, and publishing companies.

In the area of multimedia, teachers employ a wide variety of centrally produced materials including filmstrips and videos, CD-ROMs and other software, presentation graphics and even complete learning resources, such as are produced by Plato.[9]

It is important to review the “old ways” of sharing resources not only to show that resource sharing is an established fact in today’s classrooms, but also to point to some of the elements of resource sharing already in place. For it is reasonable to expect that many of the elements of resource sharing “the old way” will be replicated in an online environment.

For one thing, as mentioned, various publishers and content producers produce resources centrally and distribute them to classes around the world. And while many of these resources are distributed for free, the majority of shared resources in classrooms are purchased from their respective producers or intermediaries.

Textbook publishing and sales, especially, is a lucrative industry. The National Association of College Stores estimates U.S. / Canadian college store sales to be $8.959 billion for the 1998-99 academic year.[10]

Second, for the most part, the resources distributed in this manner are not entire classes, but rather, components of classes. This is most clearly the case for classroom aids such as wall maps and posters. But even more comprehensive materials such as textbooks are used only in part, as part of a class.

The vast majority of course syllabi require that students obtain more than one textbook. Courses frequently use only parts of textbooks; entire chapters are omitted as being beyond the scope and purpose of the course. Moreover, students frequently use parts of books (or parts of journals) in their research and reading. That’s why most university libraries come equipped with photocopiers.

Contemporary Sharing

Though most educational institutions offer only complete courses online, many other agencies have started offering smaller, more portable learning materials. These materials fall short of what we will later define as ‘learning objects’, but they offer some insight as to the direction and potential of online resources.

Immediately we see a division of the territory into, first, the learning resources themselves, and second, lists (or portals) of learning resources. In some few cases (usually where the institution has a wealth of content) the services are combined.

In Canada, the leading learning resources portal is probably Canada's SchoolNet.[11] Follow the link from the home page into "Learning Resources" and select a topic area. A list of resources is displayed, each with a short description and a link to an external website. SchoolNet also provides metadata information for each site and provides an "advanced search" using metadata.[12] Each resource in the "curriculum" area is approved by a professional "pagemaster".[13]

For the most part, however, SchoolNet lists and links to institutional home pages, and not to learning resources *per se*. Teachers using the SchoolNet service must still search through these sites in order to locate suitable materials.

Linking directly to learning resources themselves is a site based in the United States and maintained by the Educational Object Economy Foundation.[14] Merlot[15] currently lists more than 2,000 learning applications that can be accessed via the world wide web. These applications are specific materials on specific topics; for example, Merlot lists such items as "Chaucer",[16] "The Great 1906 Earthquake and Fire"[17] and RSPT Expansion (Perturbation Theory).[18] Materials are sorted into category and subcategory and have been contributed by educators from around the world.

Educators attempting to use Merlot's resources, though, will still experience frustration. Although the topic hierarchy is more detailed than SchoolNet's and although much more focused resources are listed, educators would still have to spend quite a bit of time browsing for materials. Moreover, there appears to be no resource metadata and the search mechanism provided on the Merlot site is no better than standard web search engines.

As we can see from the discussion of articulation, above, what is needed is a mechanism for connecting online learning resources with detailed course objectives. This much more advanced form of resource listing forms the basis for the selection and categorization of resources in MCI WorldCom's MarcoPolo project.[19]

MarcoPolo is a compilation of teaching resources from six educational institutions which provide free internet content for K-12 education. What the six partners have in common, and what makes this an important and interesting development in online learning, is an adherence to national curriculum and evaluation standards in the subject areas. Material is categorized by grade level and individual items are matched to individual learning topics.

Despite its strengths, however, MarcoPolo is a closed project; only the six member institutions contribute content. There is no centralized search facility and no metadata listings for the resources. The only curricula supported are United States school curricula, so the resource is not useful in a global marketplace.

Other resources are available, but these three sites typify the contemporary art of shared learning resources. Much must be done to make these resources widely useful. They need much better systems of categorization and searching. They need more robust mechanisms for updating and submissions. They need to be tied more closely to learning objectives, but in such a way as not to be tied to a specific curriculum.

An even greater weakness appears when we look at the collective set of learning resources (or applications, as Merlot calls them) offered by the three sites. It is almost not possible to identify consistency in format, scope, methodology, educational level or presentations. Some resources include lesson plans, but many others do not. Some are authored in Java, others in HTML, and others in a hybrid mixture known only to the author. Some involve ten minutes of student time, others would occupy an entire day. And there is no structured means for an instructor to know which is which.

Creating Content and The Cost of Online Learning

It may be objected that university courses are fundamentally different from K-12 courses. While there is a great deal in common between Grade 1 English from school to school, university courses are individual entities in their own right. Each time a course is offered by a university professor, it is created anew, adding a new interpretation or a new reading of familiar material.

True enough, but against that model we must look at the cost of creating courses, and especially online courses, in this manner. Creating an online course from scratch is a long, labour intensive process. Costs can vary from $4000 (all figures in Canadian dollars) to $100,000.

To cite a typical example, in *Managing Technological Change*, for example, Tony Bates estimates that a course consumes 30 days of a subject expert's time, plus an additional seven days for an internet specialist, plus additional expenses for copyright review, academic approval, and administration.[20]

A budget for course development, adapted from Bates, looks like this:

Subject Experts	30 days @ $400 / day	12,000
Internet Specialist	7 days @ $300 / day	2,100
Graphics and Interface Design	4 days @ $300 / day	1,200
Copyright Clearance		700
Total Direct DET Costs		16,000
DET overheads	25% of 16,000	4,000
Faculty of Education Approval		4,000
TOTAL		24,000

Bates is conservative. He assumes an experienced course author and HTML specialist. He does not include any instructional design costs. Course design is straightforward and does not involve the development of any interactive media or course specific Java programming. All of these would add significantly to the $24,000 total cost.

Delivery costs on Bates's model amount to an additional $13,161, as follows:

Library		1,000
Server costs		300
Tutoring	40 students @ $220	8,800
Registration	$14 x 29	406
Administration	$28.86 x 40	1,155
Printed materials and postage		1,500
TOTAL		13,161

To cover these costs, students in Bates's course pay $463 or $695 in course fees, plus an additional $177 for required readings. Students must also cover some postage and obtain access to the internet (which is provided for students working on campus). This figure is obtained by dividing the cost of offering the course over four years with an enrollment of 40 students per year over 4 years.

Almost all online course developers use the design model Bates describes. It involves a course being developed from scratch, using nothing more than a traditional university course or a good textbook as a guide. The course author typically authors *all* the content, including examples and demonstrations, quizzes and tests. Because of the cost of development, there is little use of course specific software or multimedia. The course is then offered to a small number of students over a limited time, resulting in course fees that are comparable, if not greater than, traditional university course fees.

We can do so much better than this.

We need to design online courses – even university courses – in such a way as to reduce these costs without diminishing the value of a university education. We need to do this by extracting what these courses have *in common* and by making these common elements available online.

Let me start with some examples.

Consider the Teacher's Guide to the Holocaust.[21] This site consists of dozens of resources on the Holocaust may be used and reused by any teacher approaching the subject. Each of the 'class activities'[22] could be treated as an individual learning object. The Holocaust is a very large subject - much larger than sine waves - and is appropriately divided into many components. But it is far easier, and of far greater quality, to assemble a lesson or series of lessons from these materials, than to create something from scratch.

Or consider *Hamlet*. There is not of course one single description of Hamlet, but there is only one text of the play *Hamlet* and it is not a stretch to envision a definitive online multimedia edition. Such an edition would not only contain the text, it would also contain video clips, audio clips, commentary from selected sources, pop-up glossaries, and more.

I have actually seen a CD-ROM version of Hamlet presented this way; all that is needed is online distribution.

It is not a stretch to imagine a multimedia company spending a million dollars on such a production. Assume that Hamlet is taught in 10,000 schools, colleges or universities around the world (hardly a stretch). Assume 20 students per class (an underestimate, to be sure!). At $5 per student, the company would make it's million back in one year! The economics are very good, and this excellent resource would be cheaper than even the book alone.

A course specializing in Hamlet would employ the digital Hamlet as a central resource, and incorporate as well essays, discussions and articles from scholars around the world. There is no reason why an academic journal cannot contribute a learning object (aka., article, or even a set of articles).

A "description" of the sine wave – or an account of the Holocaust, or a reading of Hamlet - becomes "a piece of learning material" when it becomes able to meet a "learning objective." Of course by 'description of a sine wave' we refer to more than merely a page or two of text plus an illustration. That's not what happens in the classroom; students are given a variety of examples, asked to calculate their own examples, are tested on their understanding, etc. A better phrasing, perhaps, is a 'lesson on sine wave functions'.

B. Learning Objects from a Theoretical Perspective

Course Construction and RAD

Courses developed along the Bates model are expensive because of two major (and related) design features: first, all course material is created from scratch, and second, this material is applied only to the limited number of students taking this particular course. In order to lower costs, therefore, a course development program is needed which enables to avoid creating everything from scratch, and to allow created course content to be applied to a much larger number of students.

From a certain perspective, an online course is nothing more than just another application, and software engineers have long since learned that it is inefficient to design applications from scratch. Educators need to learn design techniques learned by the software industry long ago, and in particular, they need to learn a concept called ‘Rapid Application Design’ (RAD).

Rapid Application Design is a process which allows software engineers to develop products more quickly and of higher quality. RAD involves several components, including a greater emphasis on client consulting, prototyping, and more informal communications.[23] But of interest here is the engineers’ re-use of software components within the context of a CASE (computer-aided software engineering) environment.

The idea of RAD for software development is that a designer can select and apply a set of pre-defined subroutines from a menu or selection within a programming environment. A good example of this sort of environment is Microsoft's Visual Basic, [24] a programming environment that lets an engineer design a page or flow of logic by dragging program elements from a toolbox.

Similar methodologies exist for a wide variety of creative or constructive tasks. A professional chef, for example, will carefully design a kitchen environment so that when he is called upon to create Crepes Suzette, the essential ingredients – including pre-mixed recipe ingredients. Auto mechanics also work in a dedicated environment and also have at hand every tool and component they may need to fix anything from a Lada to a Lamborghini.

Online course developers, pressed for time and unable to sustain $24,000 development costs, will begin to employ similar methodologies. An online course, viewed as a piece of software, may be seen as a collection of re-usable subroutines and applications. An online course, viewed as a collection of learning objectives, may be seen as a collection of re-usable learning materials. The heart – and essence – of a learning object economy is the merging of these two concepts, of viewing re-usable learning materials *as* re-usable subroutines and applications.

Educators in the corporate and software communities have known about this concept for some time. As Wayne Wiesler, an author working with Cisco Systems, writes, "Reusable content in the form of objects stored in a database has become the Holy Grail in the e-learning and knowledge management communities."[25]

Object-Oriented Design

To delve more deeply into the construction and organization of learning objects, it is necessary to introduce another concept from computer programming, object-oriented design.[26] The idea behind object-oriented design is that prototypical entities are defined, which are then cloned and used by a piece of software as needed.

Suppose, for example, as a programmer you needed to store information about 'students'. You would first design a prototypical student and define for it properties common to all students. Many aspects of the prototypical student would be undefined, however, such as the student's name, age, or phone number. These unknowns would be given placeholder values (or 'defaults') until they are defined.

When a program needs to work with a student, it refers to the prototype and 'clones' a copy of the prototype in the computer's memory (it's actually called 'cloning' in computer science – in perl the prototype is cloned and 'blessed' to reserve its place in memory). The newly cloned prototype is given a name, and then values or attributes are assigned to it. For example:

```
Clone_object: type=student id=New_student
```

```
New_student -> name = “Fred Smith”
New_student -> age = “32”
New_student -> phone = “555-1212”
```

Where object-oriented design gets interesting – and useful – is in the methodology used to *construct* object prototypes.

For clearly, an entity like Fred is a complex entity. Fred is an animal, so he has animal properties, such as age, height or weight. Fred is human, so he has human properties, such as a birthday, eye colour, and hair colour. Fred is a Canadian, so he has properties common to all Canadians, such as a social insurance number and a postal code (were Fred American, he would have a social security number and a zip code). Fred is a student, so he has student-specific properties, such as a student number or a list of classes. Were Fred an instructor, he would have instructor-specific properties as well, such as a parking spot.

When we define a student prototype for the first time, it makes no sense to define for this prototype alone each and all of these properties. This would mean that we must define similar sets of properties for all the people involved in an educational setting: students, instructors, cafeteria workers and groundskeepers.

Rather, what happens in object-oriented design is that the most basic prototype is constructed first – in this case, a generic ‘animal’ prototype. Then, the next more detailed prototype, a ‘human’ is defined. The human prototype “inherits” the animal prototype; that is, we say that all the properties an animal can have, a human can have as well. Thus, when we create the human prototype, we need only create those properties and behaviours that are *unique* to humans.

And so on up the hierarchy. When we create a ‘student’ prototype, we define a student as inheriting all the properties of a ‘human’, or all those properties of a ‘Canadian’, and define only those properties that are unique to students. Thus programmers can quickly and efficiently create a new type of entity – a special class of students, for example, or a new nationality – by inheriting the necessary properties from more generic entities.

Object prototypes also define prototypical actions or behaviours for their clones. For example, a behaviour we might expect from a student is to register for a course. The student prototype has this behaviour pre-defined as a function; when a clone is created, it comes complete with this behaviour.

Hence, we can make our clone do things by referring to these pre-defined functions (or, in computer terminology, ‘*methods*’). To have Fred Smith register in a course, for example, we would execute a command that looks something like this:

```
New_student -> register_in_a_course(course_id = “3212”)
```

The course into which Fred is registering is itself another object. In our management system, a course prototype has been defined, and at some point, a specific course has been created using that prototype. When the function ‘register_in_a_course’ is executed in Fred, the Fred-object communicates with the course-object and executes a related function in the course object, ‘add_student_to_course’.

Objects may interact – or more generally, be *related* to each other, in many ways. The most useful and common form of interaction is the *containing* interaction. Just as Fred may *contain* various other objects (such as a heart or a liver, most obviously, but also $4.95 in change, a six inch ruler and a pager), one object may in general contain one or more other objects. A course may contain students, for example. Or a course may contain units or modules. A unit may contain a test.

Each of these items is an object, defined from a prototype, which may interact with other objects in predefined ways. In a course which contained both a unit test and a grade book, for example, the unit test could interact with the grade book. What would happen is that Fred (the ‘student’ object) would interact with the test (the ‘test’ object’), which in turn would interact with the grade book (a ;grade book’ object).

Open Standards

A third major concept drawn from the world of computing science – and especially from the recent emergence of internet technologies – is the use of *open standards* in course construction.

An open standard is like a language understood and used by everyone. Just as, for example, the meanings of such terms as ‘Paris’, ‘the capital of France’, and ‘European’ are understood by almost all speakers of English, so also in an open standard are the meanings of terms and definitions widely understood and shared.

The open standard with which most online educators are familiar is Hypertext Markup Language, or HTML. This language is a shared vocabulary for all people wishing to read or write internet documents. The term ‘<h1>’ is commonly understood as a header tag; the term ‘<I>’ denotes italics.

Open standards may be contrasted with *proprietary*, or *closed* standards. Consider a document written in an older version of MS Word, for example. This word processing program used a special set of notation to define italics, bold face, and a wide variety of other features. Because other software manufacturers did not know these standards, only people using MS Word could read a document written in MS Word.[27]

The purpose of open standards is to allow engineers from various software or hardware companies develop devices and programs that operate in harmony. A document saved in an open standard could be read, printed or transmitted by any number of programs and devices.

The IMS protocols and SCORM

The IMS (Instructional Management Systems) Project is a consortium of educational institutions, software companies and publishers. The Project's objective is to

> promote the widespread adoption of specifications that will allow distributed learning environments and content from multiple authors to work together (in technical parlance, "interoperate").[28]

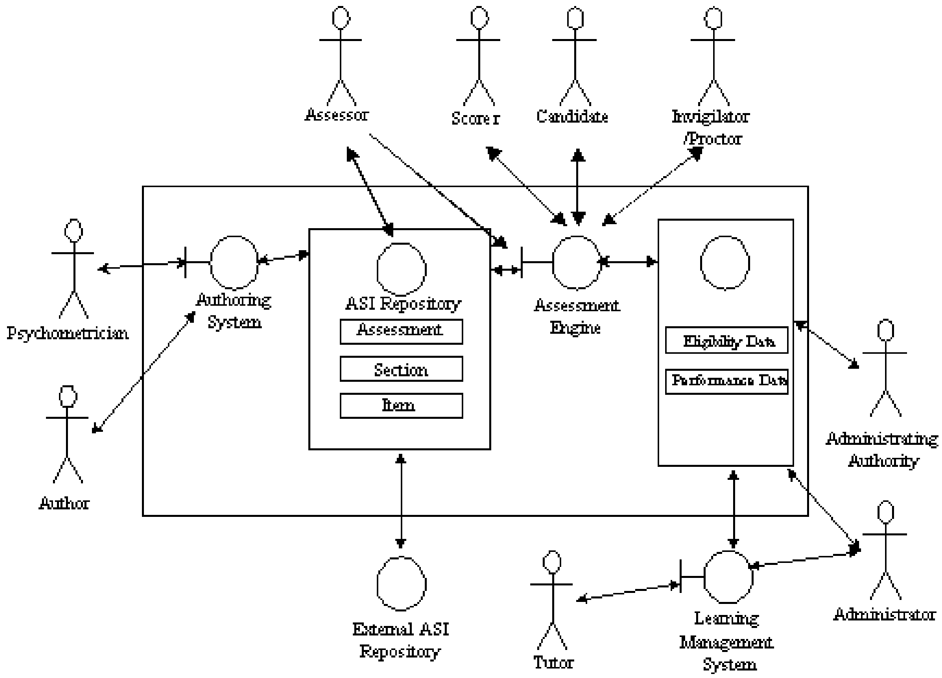
By "distributed learning environments and content", the authors mean different sets of learning materials, authored in different programming languages using different programs and located on different computers around the world.

This is an elusive goal. It amounts to enabling content produced using Blackboard[29] and stored on a computer in Istanbul – an interactive atlas, say – to be used in a course authored in WebCT[30] and located in Long Island, New York. And by 'used' what is meant in this context is that the two elements – the atlas and the course – could *interact* with each other; the atlas, for example, might report to the course *how long* a give student spend studying cloud formations, and the course might instruct the atlas to display the appropriate university logo and links to discussion boards.

In order for this to work, the atlas in Turkey and the course in the United States must define similar objects in a similar manner. For example, both programs must understand what was meant by 'course', or 'institution', or even 'logo'. Thus there is a need to obtain a common definition of the objects and properties used by the two separate systems.

Thus, the core of the IMS specification involves the definition of prototype objects (or more accurately, descriptions of prototype objects, since they would be defined differently using different computer languages). The IMS Enterprise Information Model,[31] for example, defines a 'Person Data Object', a 'Group Data Object', and a 'Membership Data Object'.

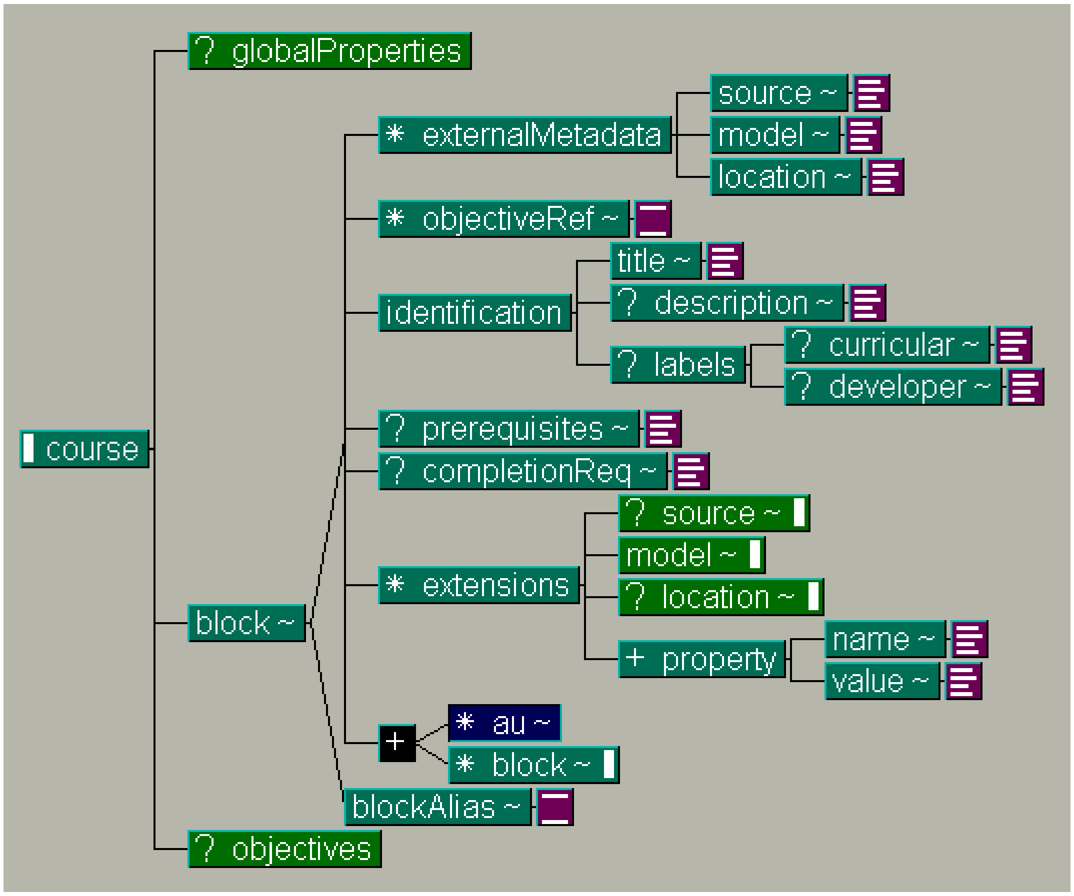
In a similar manner, objects must interact with each other in predefined ways. If one program is expecting a grade as a digit and calls it 'grade', and the other sends it as a word and calls it 'score', then the two programs are unable to interact. A document like the IMS Question & Test Interoperability Information Model Specification[32] defines the manner in which various components of a testing system interact with other elements of a wider instructional management system. The figure below, drawn from the same document, is illustrative of the interactions being considered:



The diagram depicts the types of objects which interact. The little stick men are person-objects, and it is worth noting that no fewer than nine separate types of person-object are defined. The circles represent "key components", each of which is an independent piece of software: the authoring system, for example, or the assessment engine.

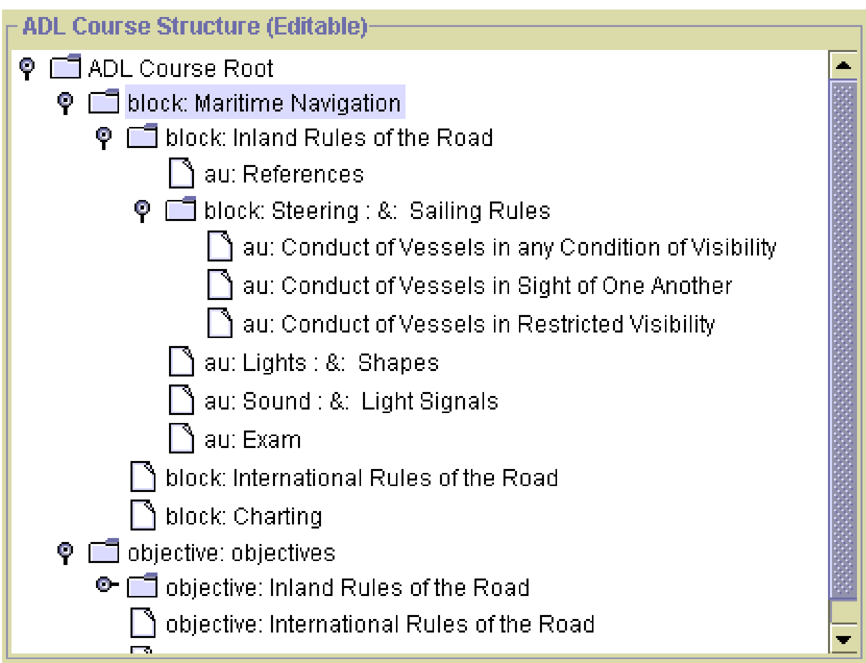
More detailed implementations of this basic structure are defined by more specific projects. One major project of this type is Advance Distribute Learning's Sharable Courseware Object Reference Model, SCORM.[33] This document describes in detail the object hierarchy in a course and how objects' methods (which are, recall, predefined functions) are defined.

Here is a sample hierarchy from SCORM, displaying only the 'Global Properties' node:[34]



The *globalProperties* node contains or references information about the course as a whole, such as prerequisites and course identification. It also provides information describing the general approach used during the design of the course.

Drilling more deeply into the course itself, SCORM defines the course components:



This diagram defines some of the major components of a Maritime Navigation course. Note how the typical course components, such as references, exams, and lesson objectives are each included is distinct components. Each of these elements is an object in its own right; the course as a whole – also an object – contains these discrete objects.

As we can see, the IMS Protocols, and specific implementations, such as SCORM, define in detail the potential structure of an online course. In IMS and SCORM, a course – and the elements surrounding a course, such as students, grade books, and prerequisites – are depicted as interacting and inter-related objects.

A Common Language

Thus far we have considered only what may be called the *semantics* of a learning object economy. We have looked at what things (or objects) there are, what they do, what we call them and how we define the meanings of our words.

As yet, we have not considered how such a system might be *distributed* and *interoperable*. We have not show just *how* a computer system in Turkey might send information to a program running in the United States. In order for these two systems to communicate, it is important not only that they be talking about the same *things* but also that they have a common *language*.

The common language adopted by IMS and SCORM – and being adopted by database programmers, librarians and designers around the world – is the eXtensibe Markup Language, or XML, developed by the World Wide Web Consortium.[35]

XML is a means of representing documents according to their internal structure. Thus, for example, a book containing chapters and verses might be represented in XML as follows:

```
<tome name=”Bible”>
        <book name=”Genesis”>
                <chapter name=”1”>
                        <verse name=”1”>
                                In the beginning God created the heaven and the
                                earth.
                        </verse>
                        <verse name=”2”>
                                And the earth was without form, and void; and
                                darkness was upon the face of the deep. And the
                                Spirit of God moved upon the face of the waters.
                        </verse>
                        …
                </chapter>
                …
        </book>
        …
</tome>
```
[36]

In a similar manner, a course containing units, modules and exercises may also be represented in XML.

```
. . .
<course>
   <block  id="B1">
         <identification>
            <title>Maritime Navigation</title>
            <labels>
               <curricular>UNIT</curricular>
            </labels>
         </identification>

         <block id="B2">
            <identification>
               <title>Inland Rules of the Road</title>
               <labels>
                  <curricular>MODULE</curricular>
               </labels>
            </identification>
            <au id="A1">
               <identification>
                  <title>References</title>
               </identification>
               <launch>
<location>/Courses/Course01/Lesson01/au01.html</location>
               </launch>
            </au>
            <block id="B3">
               <identification>
                  <title>Steering &#38; Sailing Rules</title>
                  <labels>
                     <curricular>MODULE</curricular>
                  </labels>
               </identification>
```



```
                      . . .
```

XML has two useful features in the context of the current discussion:

First, it is *structured*. An object hierarchy may be defined such that one object may contain other objects, and such that any given object may be assigned any number of properties. Thus, XML is capable of representing an object hierarchy as defined above. All the components of our online course object economy may be described in XML, even non-digital objects such as students, classrooms and books.

And second, it is machine-readable (and machine-writeable, which amounts to the same thing). This means that a computer program can *produce* a properly formatted XML document using information stored in, say, a database, and it means that another, different, computer program can *read* that file and assign the proper values to the proper variables in its own internal representation system.

XML is to structured information what HTML is to structured documents. Each provides a means of distributing content to other systems no matter where they are located and no matter what program they are running. Thus, a piece of learning material, no matter where it is located, may be seamlessly integrated into an online course, provided that the XML tags are employed consistently – that is, provided the semantics are the same.

In an XML document, a schema establishes the semantics of a system of tags.[38] For example, the Dublin Core establishes a schema for referring to printed documents.[39] Any XML document which describes a book (and which uses Dublin Core) would use XML tags (and hence, assign corresponding properties to a corresponding book object) defined by the Dublin Core Metadata Element Set.[40]

C. Authoring Learning Objects

Authoring Learning Objects - Data

While today most guides and references discuss online *course* authoring, the proper reference point is the authoring of learning objects, where a learning object is an element of a course as described above. As we have seen, a learning object may be one of any number of items: a map, a web page, an interactive application, an online video – any element that might be contained inside a course.

There are two major facets to authoring learning objects: first, the content of the learning object itself, and second, the *metadata* describing the learning object. We might think of authoring learning objects as akin to authoring pieces of a puzzle, in which case the content is the image or picture on the surface of the piece, while the metadata is the shape of the piece itself which allows it to fit snugly with the other pieces.

Today by far the most common medium for content is hypertext markup language (HTML). Course authors are able to employ a variety of authoring tools such as Microsoft's FrontPage[41] or Macromedia's Dreamweaver.[42] These tools enable the creation of quite sophisticated pages, especially FrontPage, which through a series of extensions allows authors to embed interactive applications into the page.

The problem with these HTML pages is that they're not portable, especially not FrontPage generated files, which must interact with a Microsoft server. A web page designed for one course at one university will contain course and university specific information: the name of the course, the name of the university, and even a colour scheme. To be used or adapted by another course, the pages need to be redesigned.

Moreover, HTML pages – especially pages designed using FrontPage – do not display well in multiple formats. A separate version must be created if, say, the page needs to be delivered over wireless access protocol (WAP)[43] or if it is input as data for analysis by a Javascript or CGI process. HTML – as it is currently implemented by these products – combines content and presentation information, thus narrowly limiting its portability.

In order to be portable, a document's content must be, first, structured, and second, separated from presentation information. This goal is accomplished by XML, which uses tags to structure information and which refers presentation information to a separate document entirely (an XSL file; see below).[44]

A significant step in the right direction is to create course materials *not* in HTML, but rather, in a structured markup language such as XML. A good example of this is the approach taken by British Columbia's Open Learning Agency, which creates its courses in SGML (Standard Generalized Markup Language, a tagged language very similar to XML).[45] By organizing content in this way, print versions, web versions or even wireless versions may all be produced from the same base document in a matter of seconds. Structural elements such as tables of contents and page numbers are generated on the fly, while course or institution specific information is defined in the template. And specialized documents, such as course outlines, amy be generated from the same source.[46]

SGML documents may be generated and edited using any common SGML editor.[47] But the implementation – at least as used by OLA – is not portable. The course documents are undifferentiated wholes, so they would have to be adapted by other institutions *as a unit*.

In any case, it is not reasonable to employ one language for all parts of an online course. What we are more likely to see – and are beginning to see – is a set of different languages for different parts. IMS is slowly drafting these specifications and now has four sets: meta-data, enterprise, content packaging, and question-and-test.[48] Related sets of specifications are being defined by the World Wide Web Consortium, such as Math Mark-Up Language (MML)[49] and the Synchronized Multimedia Integration Language (SMIL).[50]

Rather than use a single tool, such as an XML or SGML editor, course authors will begin to use tools designed for specific purposes. Already, we have seen some of these developed, one of the most popular being Half-Baked Software's Hot Potatoes, a tool for designing online quizzes[51] (it is worth noting that the next version of Hot Potatoes will produce XML-XSL based output). It is not hard to image a suite of standards-compliant applications emerging into the marketplace: one for drafting course outlines, one for creating individual lessons,[52] another for authoring slide shows, another for creating case studies, and so on.

For example, the University of Bristol's TML (Tutorial Markup Language)[53] described a common authoring language for online tutorials and quizzes. The purpose of TML is to "designed to separate the semantic content of a question from its screen layout or formatting" and in so doing, provides a structural framework for tutorial content (the boxes are not part of the document, and are placed there for clarity).

```
 -----------------------------------------------------------------
|<!DOCTYPE TML PUBLIC "-//ETS//DTD TML 4.0//EN//" [ ] >           |
|<TML>                                                            |
| <-- Arbitary normal HTML -->                                    |
| |-------------------------------------------------------------| |
| |<TUTORIAL>                                                   | |
| |<QUESTION ATTEMPTS=3 NAME=Capitals TYPE=Multiple-Choice>     | |
| | |--------------------------------------------------------- | |
| | |<p>The text of the question. It consists of HTML text.</p>| | |
| | |<CHOICES>                                                | | |
| | | |-----------------------------------------              | | |
| | | |<CHOICE CORRECT>This is a correct choice |             | | |
```

```
| | | |<CHOICE>This is an incorrect choice      |              | | |
| | | |     .                                 |              | | |
| | | |     .                                 |              | | |
| | | |----------------------------------------              | | |
| | |</CHOICES>                                              | | |
| | |<SCORE>                                                 | | |
| | | |----------------------------------------              | | |
| | | |<GAIN CORRECT ATTEMPT=1 VALUE=3>       |              | | |
| | | |<GAIN CORRECT ATTEMPT=2 VALUE=1>       |              | | |
| | | |<LOSE HINT VALUE=1>                    |              | | |
| | | |     .                                 |              | | |
| | | |     .                                 |              | | |
| | | |----------------------------------------              | | |
| | |</SCORE>                                                | | |
| | |<HINTS>                                                 | | |
| | | |----------------------------------------              | | |
| | | |<HINT>This is a hint                   |              | | |
| | | |<HINT>This is another hint             |              | | |
| | | |     .                                 |              | | |
| | | |     .                                 |              | | |
| | | |----------------------------------------              | | |
| | |</HINTS>                                                | | |
| | |<RESPONSES>                                             | | |
| | | |----------------------------------------              | | |
| | | |<WHEN CORRECT><B>That's right!</B>     |              | | |
| | | |<WHEN OPTION=d>You were close that time |             | | |
| | | |<WHEN INCORRECT>Sorry, that was wrong  |              | | |
| | | |     .                                 |              | | |
| | | |     .                                 |              | | |
| | | |----------------------------------------              | | |
| | |</RESPONSES>                                            | | |
| | |----------------------------------------------------------- | |
| |</QUESTION>                                                 | |
| |<QUESTION ATTEMPTS=3 NAME=Protocols>                        | |
| |     .                                                      | |
| |     .                                                      | |
| |                                                            | |
| |</QUESTION>                                                 | |
| |------------------------------------------------------------| |
|</TUTORIAL>                                                     |
|</TML>                                                          |
|----------------------------------------------------------------|
```
[54]

What is interesting about the TML project is that software have been developed both for authoring and for displaying TML documents.[55] Demonstrations available online, such as Crisp and May's Chemistry tutorial,[56] show how a TML file would be rendered as a series of HTML pages viewed by the student.

Authoring Learning Objects – Multimedia

The model for most of the learning materials described above – authored by a subject mattered expert, presented in text (even with supporting graphics and animation) – is the book, or at the very least, the course manual or course guide.

More and more non-textual resources are appearing every day, however. Video clips, small applets, interactive animations, simulations – these are authored using a wide variety of programs ranging from video editing software to Java editors to Macromedia's Director.

Many of these are available online, such as the animated slide show, "Deepest Impacts: A Species Demise E.L.E."[57] They are developed and distributed because, as J. Bradford

DeLong, a developer of several economics animations, writes, “I think that there is a reasonable chance that [they] are--or could become--a vast improvement over the textbook presentation.”[58]

Many more resources are not available online. Schools face continual pressure to purchase a wide variety of educational CD-ROMs and teaching software.[59] Thus even online courses present challenges for students and instructors as various software applications need to be purchased, delivered and installed into students’ computers.

With the emergence of Applications Service Providers (ASP – not to be confused with Microsoft’s Active Server Pages) the distribution of software via CD-ROM and floppy disk will slowly evaporate. Application Service Providers are online services that automatically deliver and install software on an as needed basis to client computers.[60]

Designing Learning Objects – Data or Multimedia

The sections above have described the mechanics of creating learning objects. Before continuing with the technical description of course components it is important to look at the *content* of learning objects.

While no doubt there will be much debate regarding the instructional design of learning objects, in practice designers have opted for a performance based or competency based theory of design.

For example, Cisco System’s RIO (Reusable Information Objects) project is explicitly performance based. Drawing on work by Ruth Clark,[61] RIO “views all training as a means to enable a worker to successfully complete a task.”[62] The process follows three steps:

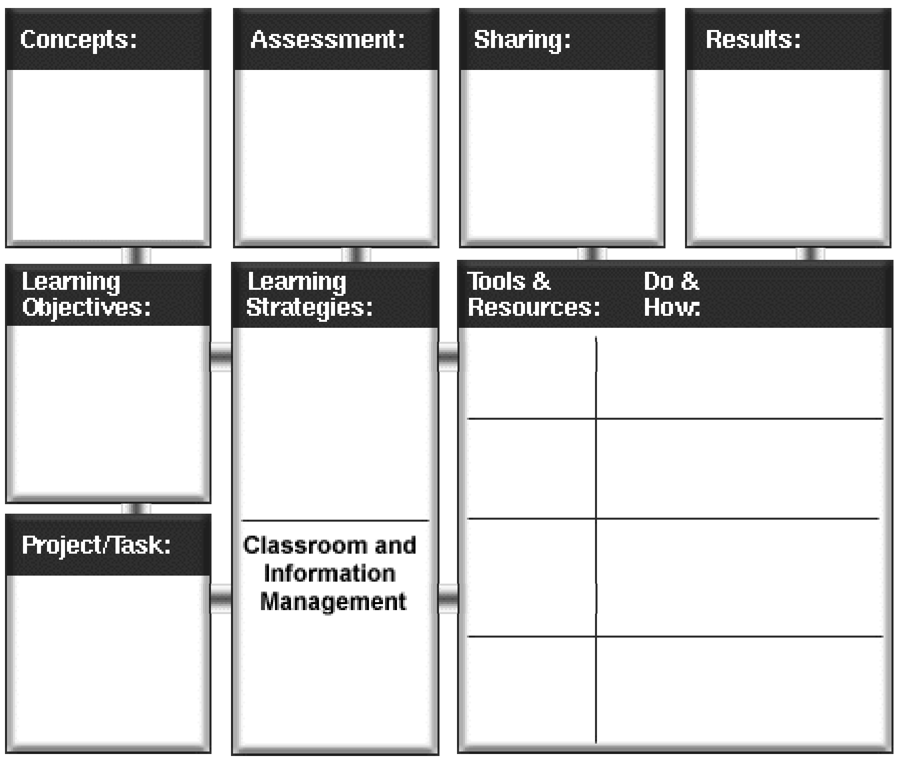
1. Identify the job task
2. Identify the skills and knowledge necessary to complete the task
3. Develop training in modular chunks that are organized to support the task

Learning, on this model, is outcome based rather than content based. It focuses on what people want (or need) to do, rather than on what there is to know.

Suppose, for example, Cisco introduced a new product. A traditional approach to training would be to list the product’s features, to develop the course based on this features list, and to test students on their recall of the features. A performance based approach, by contrast, would begin by assessing customer requirements. These requirements would then be matched with product capabilities. Students would be tested on their ability to recommend the product in appropriate situations.[63]

Most educational institutions would find a definition of learning objects based on specific tasks to be somewhat limiting. However much work has been done regarding the definition of learning *outcomes* in general, and a wider definition of learning objects would be tied to these outcomes. Specifically, the content of a learning object would be derived from a discussion of a course's (or a lesson's) learning objectives, where the achievement of these outcomes can be measured in terms of students' performance.

In sum, the overall content of a learning object would be similar in scope and nature to the content of a typical lesson. Many lesson-planning aids exist; the following is typical[64]:



This template, from Ohio Schoolnet, is notable because an instructor may click on any given component to view a detailed description. A learning object authoring environment would employ a very similar interface, while clicking on the component area would enable an editing screen for that component. Thus, for example, if the author clicked on "Learning Objectives", she would be greeted with a list of learning objects appropriate for that course, from which she would select one or more. Or if she clicked on "Tools and Resources" a list of suitable online resources would be displayed.

Authoring Learning Objects – Metadata

For any object, text-based or multimedia, an associated set of metadata needs to be created. The type of object determines the content of the metadata. For example, an image might have a property labeled 'photographer', and a piece of text might have properties labeled 'editor' or 'publisher'.

Whatever the properties, the authoring of metadata itself will be straightforward for most course designers. Because metadata files are machine-writable, authors will simply access a form into which they enter the appropriate metadata information. The form – generated either by a web page or by a specific piece of application software – will send field information to a metadata page editor.

The process of converting form data to XML data is very simple. Here is the code, in perl (assuming forms data has been saved in a standard hash file %FORM):

```
while (($fk,$fv) = each %FORM) {
       $output .= "<$fk>$fv</$fk>\n";
}
print $output;
```

More complex metadata editors will include mechanisms for parsing and displaying existing metadata documents. They will also include forms for a wide variety of resources – the list of fields in these forms are defined by schemas, as discussed above.

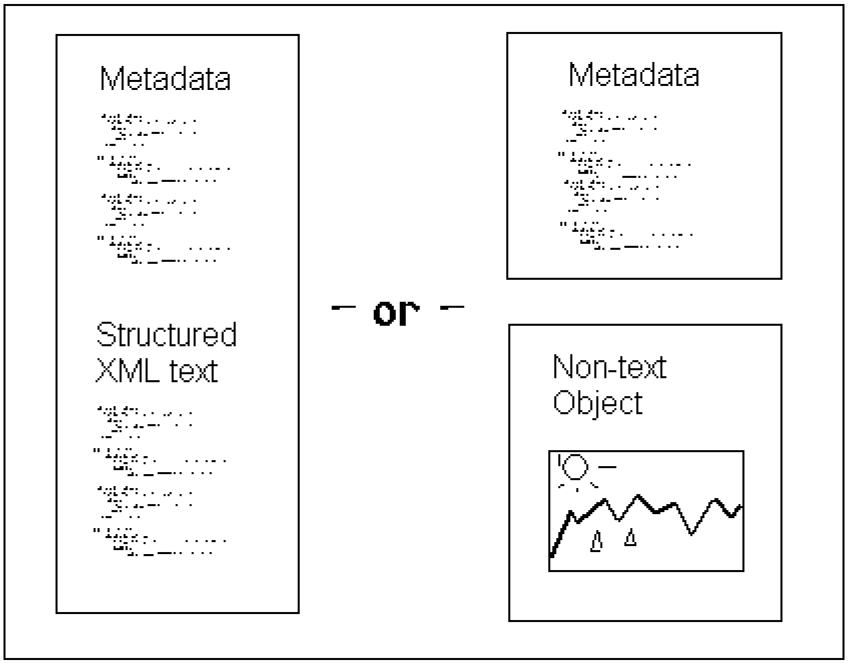
Sophisticated metadata editors will not define the fields for different types of forms internally. Rather, they will access schemas from various sources around the internet. A list of available schemas for online learning is provided on the IMS website.[65] The editor will retrieve the titles of these schemas from a central index, and once the author selects a title, will read the specifications and create the form accordingly.[66]

What is significant is that all of this occurs behind the scenes. All the author needs to know is what *type* of metadata is being created, and that type is defined by the type of object being described. As a side note – it is worth nothing that schemas for a wide variety of entities, and not just course components, are being defined in this way. See http://www.schema.net/[67] for more information.

In the case of multimedia objects, many editors will have metadata generators built in. This is already the case with some Microsoft products, such as MS-Word, which saves MS Word files in a (Microsoft specific) XML format. Such products will save users the time and trouble of typing the same information over and over (such as their name, institution, and the date).

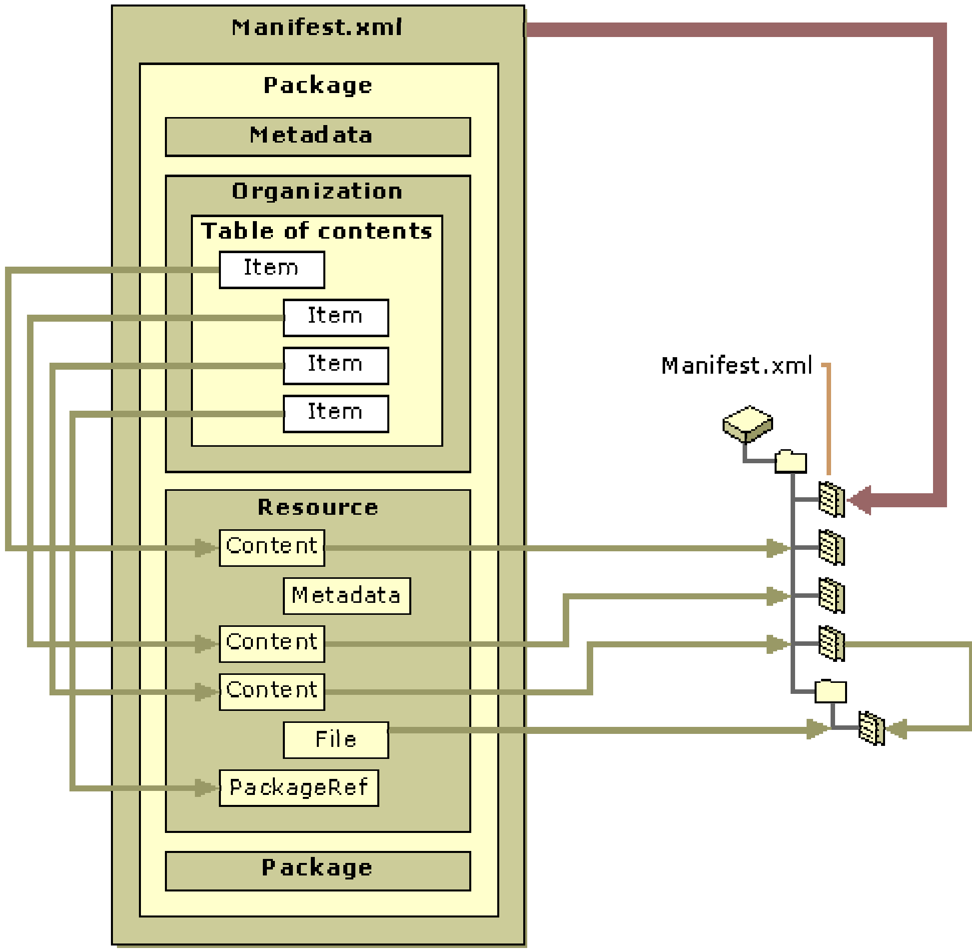
Authoring Complex Objects

By now you should have the following picture of a learning object in your mind:



Each of these objects is created and stored in a database. The contents of this database are available to course authors. Some databases may be available over the internet, while other databases will be available only internally.

In order to create a more complex entity, like a lesson, a number of these entities are collected together in what is called a package.[68] A package is a structured representation of a set of independent objects. Microsoft's LMS concept provides us with a good illustration:



69

The ‘manifest’ is like the ‘shipping label’ for the package, detailing the contents of the package. The table of contents is an ordered representation of the titles of each item. The metadata for items themselves may be actually contained in the package, or pointed to by a line in the page. Similarly, resources themselves may be contained in the package, or pointed to by a line in the package (obviously, non-textual resources, such as images, must be pointed to).

Again, the package is described in XML. However, it is unlikely that course authors will write this file by hand, or even that they will ever view the file directly. Course authors will generate this file by interacting with it through a program such as Microsoft’s LMS (hopefully *not* LMS, because it forces users into a Microsoft-only environment, thus defeating the interoperability requirement described above).

How would this work. At this point, much of what follows is speculation, since the required systems have yet to be constructed.

Using an authoring tool, an author will select (from a drop-down list) a packaged-sized entity, say, ‘Lesson’. The authoring tool will retrieve the the schema for ‘Lessons’ either from a local database or – better – from a central schema resource online. The schema defines the fields which must be filled out (filling some automatically, especially if the lesson is part of a large project).

Additionally, since the object in question is a *package*, the program knows that it will be composed of other objects: an interactive display, say, or a movie, or some other resource. These options are presented to the author: the author selects ‘insert’ and then selects the type of object to be inserted.

At this point, in traditional course authoring, the author would start to write content for the new component. And this will still be an option – if the author selects ‘new’ the appropriate authoring tool will be opened and the author can create a new resource, as described above. But many authors will select from a list of available resources.

If the author is authoring a lesson, the course authoring system already has some significant information. It knows, for example, what the topic of the course is, what the grade level is, what the geographic region is, and more. These were all defined when the course was created, and these values are inherited by any object which forms a part of the course.

If, then, the author wishes to add a resource, the authoring system has the information it needs to conduct a highly selective search of resources. The system may search a local database, but more likely, it will search an online learning objects repository. Such a repository won’t actually *contain* these resources – they will be distributed on websites around the world – but it will contain information *about* those resources. Specifically, it will contain those objects’ *metadata*.

The authoring system consults the repository and runs a search. The results of the search are provided in a menu for the course author. Some of these are approved by standards bodies, and some are not. Some are defined by grade level and even learning objective, as defined by, say, the Western Canada Protocols, and some are not. Some are available for free, while others will require that royalties be paid. The author can instruct the authoring tool to *only* accept resources approved by a certain standards body or meeting a certain learning objective, or falling within a certain price range.

The author at this point may preview the material, or she may decide to insert it into the course. At this point, the *metadata* – not the object itself – is inserted into the course package. The author moves on to the next item in the Lesson, and in a very short time – hours, not days, completes the lesson, and eventually, the course.

D. Displaying Learning Objects

Learning Object Repositories

Consider the impact of a resource like Martindale’s Health Science Guide, a resource center listing 60,000 teaching files and 129,000 medical cases.[70] Such a resource, if made available to medical schools around the world, would greatly facilitate the creation of courses in medicine and could provide a sustaining source of revenue for the Martindale Centre.

The core of a learning object repository is the central database containing the tens or hundreds of thousands of individual objects. Such databases will be multi-functional; online courses constitute only one of the end uses to which these objects will be put (other uses might include online journals and magazines, personal websites, knowledge management applications, and more).

Often, these databases will be working databases for separate enterprises entirely. For example, a government may place all legislation, regulations, procedures manuals and tables into a database. This information would be accessed by an array of applications and end users, including lawyers, real estate agents and the press. The very same set of resources would also be made available to online courses.

Attached to each object in the database will be a metadata file, as described above. This file will include subject-specific information, but also, information as it applies to online learning (such as grade level, subject area, and more). The cost structure for materials retrieval will also be included in the metadata.

A system of learning objects repositories around the world (it is very unlikely that there will be one) will access this metadata to form its own, compiled, set of resources. The online learning repository will retrieve only that metadata relevant to online learning. It is this, filtered, metadata that will be accessed by online learning systems.

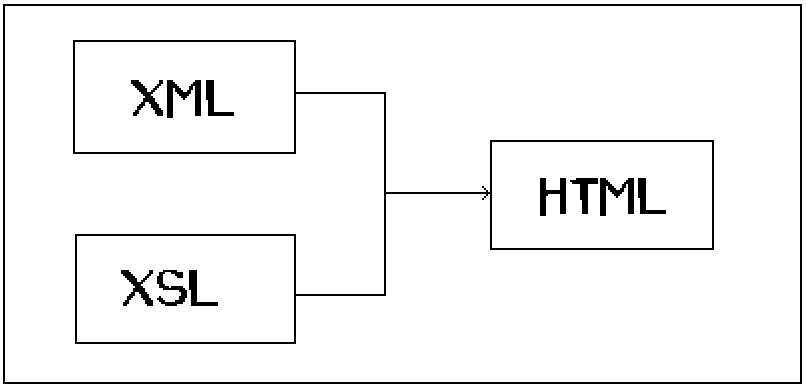
Such a system may seem far-fetched but is already being implemented in online journalism. Content producers, such as Reuters, provide their material to content syndicators, such as I-syndicate[71], Individual.com,[72] or Netscape Netcenter[73]. These sites retrieve the content – in the form of Rich Site Summary (RSS)[74] or other XML-type files, process it for display, and relay it to individual users. Individual users, playing the role of 'newspaper editors', can create customized daily newsfeeds which appear on a web page or in their email every day.

Existing learning portals, such as Learn2.com, HungryMinds, Learn.com, Fatbrain, and SmartPlanet, are beginning to move toward this model of content delivery.[75] Topic specific business-to-business learning portals are providing customized learning from within the context of learning management systems. TrainingTek.Com, for example, allows course designers to select learning objects from a menu of options within the context of their learning management system.[76] A similar resource was recently launched by internet and publishing giant, America OnLine.[77]

Displaying Learning Objects

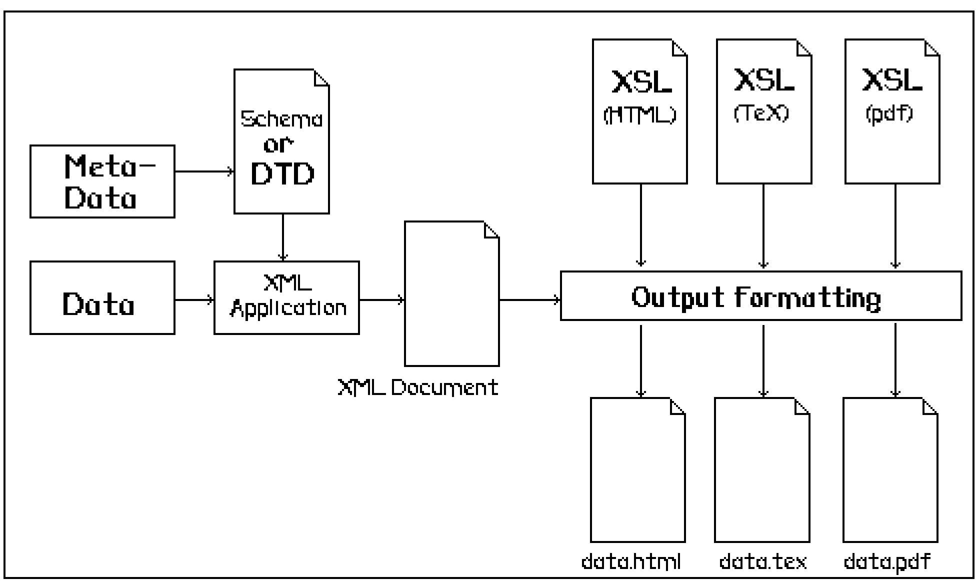
Because learning objects are distributed as XML files, they may be displayed using a wide variety of hardware and software combinations.

The most simple and straightforward implementation of this is through the conjoining of XML files with related style sheets defined in XSL (extensible Stylesheet Language), as mentioned above. For example, an XML file and an XSL file merge to create an HTML file:



In this simple example, each element in an XML file is associated with an output format (defining such things as font styles and sizes or background colours) in the XSL file. The XSL file describes these elements in standard CSS (Cascading Style Sheet).[78] Thus combining the XML and XSL definitions yields an HTML output understood by web browsers.

Because style and substance are separated, the same XML data may be output in a variety of formats:



The learning object itself is composed of data and metadata (left), stored in a database. An XML application interprets the data through a schema, which defines the properties that will be displayed in the resulting XML document. This document is sent to output formatting, which will use one XSL file to produce an HTML page, another XSL file to produce a LATEX page, and yet a third XSL file to produce a printable PDF page.[79]

This model provides tremendous flexibility. For a given set of data, one set of metadata may produce an XML file useful for online students while another may produce an XML document useful for real estate agents, depending on which schema is used. And any given XML document may be output in a variety of formats, depending which XSL file is used. One XSL file may define the format used by the University of Chicago while another may define the document template used at the University of Toronto. One style sheet may be used for online viewing, while another may define the print version of the document.

Indeed, an XML file, merged with other XML and XSL sources, may be displayed in a highly customized or personalized format. Proposed, for example, is an agent-based learning system that recognizes individual users and formats pages accordingly. This agent would alter not only display preferences, but would also amend content according to previously established user preferences.[80]

In traditional education, learning objects would be distributed in the context of a course; that is to say, an online course would consist of an ordered collection of related learning objects unified by some set of common learning objectives or course-wide assessment techniques, such as a final essay.

Learning objects would also be used in a wide variety of non-traditional educational scenarios. Consider this proposal to the corped (corporate education) mailing list:

> Now, imagine that corporate educators build repositories of chunked learning objects (knowledge management databases), incorporate browser and search engine components, and place this electronic resource on every employees' desktop. Suppose that corporate educators use event notification to send friendly e-mail messages to users to offer assistance or to identify early warning signs of trouble (e.g., users searching for learning objects on discrimination, sexual harassment, career counseling, or abusive bosses). Imagine constructing the application to permit self-assessment tools for users, electronic registration for classes, and asynchronous and synchronous collaboration tools (e.g., chat rooms, listserves, and Web conference rooms). Corporation could even link communication technologies to permit employees to have immediate access to professional mentors. This combination of technologies could comprise a total learning solutions with the flexibility to accommodate diverse styles of learning.[81]

No matter how learning objects are distributed, their method of access will be similar in every event. By clicking on a link (or making a similar selection), a student will from his or her desktop browser send a request to the object, which will be delivered as an XML file. The request will also refer the distributor to another XML file, which will contain user specifics, such as their name, institution, and course.

On the distributor end, the learning object will be packaged and send to the client. In many cases, the package will be a simple XML file, though often, the XML file will be compiled on the fly in response to the user data sent with the request. The distributor will also note the request in its logs, update its billing or license data, and if requested, send notification to the student's employer or educational institution.

The object, when retrieved, is then fed to a viewer. The viewer may be included in standard web browsers (as, for example, graphics are today) or may require a seaparate viewer. Viewers may be defined (and even available) through the institution's web site. Most likely, specialized viewers will be downloaded and installed on an as-needed basis as java applets are today.

Different viewers will be used for different types of output device. This, just as graphics today may be viewed on a screen, send via fax, printed on pager or sent as an email attachment, so also different viewers for each type of learning object will be available for different distribution mechanisms.

The Learning Object Economy: A Polemic

We can be visionary. We can imagine a proliferation of cottage industries involved in the production of learning objects. Standards bodies, reviewers and other filter mechanisms will become important. Because a payment scheme is built-in, the model becomes sustainable. But because each individual object view is so inexpensive, online learning becomes affordable.

Yet what about traditional university education, where professors see their courses as unique creations which re-make the field of enquiry each time they are taught?[82]

This approach is the core of traditional liberal arts education. It is this very aspect of online learning which pits computer-assisted learning, such as is envisioned in a learning object economy, against traditional face-to-face professorial learning.

Let me grant that this sort of re-examination of the material is necessary and desirable. But let me question whetherthis process at the same time serves as an effective teaching methodology.

To put the question in as sharp a light as possible: do first-year engineering students need a *brand-new* Shakespeare course, or will the interpretation developed last year (or two years ago, or in Saskatchewan) do the job?

And moreover: is it fair to require that students, whose primary goal is at best a surface understanding of Hamlet, to *pay* for the development of a *brand-new* interpretation, which last year's, or Saskatchewan's, would have done just fine?

I agree that hand-rolled bread, carefully prepared by a master chef, is superior in quality to a standard loaf purchased at Safeway. But to a person who is merely hungry - rather than a connoisseur - the obligation to purchase *only* hand-rolled bread is more than just an imposition, it amounts to a denial of basis sustenance for many.

The question is: *could* we teach first-year English using 'Hamlet modules'? *Could* we reduce the cost of such learning by an order of magnitude? Are the endless creations of professors *necessary* for the eventual goal of cultural literacy? Is it reasonable to *deny* such an education to many (especially in less developed nations) in order to generate each course anew each year in each university classroom?

Sorry about all the italics, but I am trying to emphasize how it looks from the other side of the equation. And I'm trying to express the sort of thinking when such object-based courses are inevitably accredited. How will the hand-craft institutions justify their art? Sure, we need reinterpretations of Shakespeare, but do we need a thousand such reinterpretations a year?

There is very much a tension, between those who create the knowledge, and who jealously guard their monopoly over its propagation and distribution, and those who must consume that knowledge to get a job, to build a life, to partake fully in society.

My personal belief is that arts and humanities professors - even those who teach senior courses - will have to redefine their approach or be priced out of existence. Probably history, not argument, will show whether this belief is well founded.

---

[1] Distance Learning in Higher Education. CHEA Update Number 2. Council for Higher Education Accreditation. June, 1999. Last updated, December, 1999. Available http://www.chea.org/Commentary/distance-learning.html (viewed, may 1, 2000).

[2] TeleCampus. Website. Available http://courses.telecampus.edu/. Viewed 01 May, 2000.

[3] The Web of Asynchronous Learning Networks. Website. Available http://www.aln.org/. Viewed 01 May, 2000.

[4] Or so I argue. Downes, Stephen. Hungry Minds: A Commentary on Educational Portals. *Online Journal of Distance Learning Administration*, Volume III, Number I, Spring 2000 (State University of West Georgia, Distance Education Center). Available http://www.westga.edu/~distance/downes31.html

[5] Baccalaureate Core Course Equivalency. Oregon State University. Updated as of 06-Apr-2000 for Spring 2000. Available http://www.orst.edu/Dept/admindb/arttable/scr1140_arttab.htm. Viewed 01 May, 2000.

[6] Illinois Mathematics and Computer Science Articulation Guide. Prepared by IMACC-ISMAA Joint Task Force, May 1995. Available http://www.imacc.org/articulation/. Viewed 01 May, 2000.

[7] Telecampus Course Directory. History. Available http://courses.telecampus.edu/subjects.cfm?category=12. Viewed 01 May, 2000.

[8] Classroom Aids. Website. Astronomical Society of the Pacific. Available http://www.aspsky.org/catalog/class.html. Viewed 01 May, 2000.

[9] Plato. Website. Available http://www.plato.com/. Viewed 01 May, 2000.

[10] National Association of College Stores. Information You Can Use. Website. http://www.nacs.org/info/college_market/overview.asp. Viewed 01 may, 2000. Statistics cited from NACS, *College Store Industry Financial Report 2000 Edition*. See http://www.nacs.org/info/products/productinfo.asp?PRODUCT_ID=2079-00 for information.

[11] Canada's Schoolnet. Website. Available http://www.schoolnet.ca/. Viewed 02 May, 2000.

[12] SchoolNet. Advanced Search. Website. Available http://www.schoolnet.ca/home/e/search/advanced_e.asp. Viewed 02 May, 2000.

[13] Schoolnet. Pagemasters. Website. Available http://www.schoolnet.ca/pagemasters/e/. Viewed 02 May, 2000.

[14] Educational Object Economy Foundation. Website. Available http://www.eoe.org/. Viewed 02 May, 2000.

[15] Merlot. Website. Available http://www.merlot.org/. Viewed 02 May, 2000.

[16] President and Fellows of Harvard College. Geoffrey Chaucer. Last Modified: 02/07/98. Available http://icg.fas.harvard.edu/~chaucer/index.html. Viewed 02 May, 2000.

[17] Museum of the City of San Francisco. The Great 1906 Earthquake and Fire. Website. Undated. Available http://www.sfmuseum.org/1906/06.html. Viewed 02 May, 2000.

[18] Sergeev, Alexi. On-line calculation and graphical display of Rayleigh - Schrödinger perturbation series for various quantum-mechanical problems. Undated. Available http://www.bgu.ac.il/~sergeev/perturb.htm. Viewed 02 May, 2000.

[19] MCI WorldCom. MarcoPolo. Website. Available http://www.wcom.com/marcopolo/. Viewed 02 May, 2000. See also Downes, Stephen. Spotlight Site. *Technology Source*. June/July, 2000. (forthcoming).

[20] Bates, Anthony. *Managing Technological Change*. San Francisco: Jossey-Bass Publishers, 2000. pp. 138, 144. Not available online.

[21] Korczak, Janusz. Teacher's Guide to the Holocaust. Web site. Available http://fcit.coedu.usf.edu/holocaust/. Viewed 23 May, 2000.

[22] Ibid. Class Activities. Available http://fcit.coedu.usf.edu/holocaust/activity/activity.htm. Viewed 23 May, 2000.

[23] For more information on RAD, see University of California, Davis, Application Development Methodology: Rapid Application Development (RAD), April 16, 1997. Available http://sysdev.ucdavis.edu/WEBADM/document/rad_toc.htm. Viewed 03 May, 2000.

[24] Microsoft. Visual Basic. Web site. Available http://msdn.microsoft.com/vbasic/. Viewed 03 May, 2000.

[25] Wieseler, Wayne. RIO: A Standards-Based Approach for Reusable Information Objects. Cisco Systems White Paper. Undates, p. 4. Available http://www.coursenet.com/html/products/white_paper_RIO.htm in pdf format. Viewed 09 May, 2000.

[26] One of many descriptions may be found in Montlick, Terry, What is Object-Oriented Software? Software Design Consultants. 1995-1999. Available http://www.catalog.com/softinfo/objects.html. Viewed 03 May, 2000.

[27] Of course, this is not strictly true; with Microsoft's cooperation, vendors could create translation engines which would 'import' MS Word documents – but always with a loss of formatting.
[28] IMS Global Learning Consortium, Inc. About IMS. 2000. Website. Available http://www.imsproject.org/aboutims.html. Viewed 03 May, 2000.

[29] Blackboard. Website. Available http://www.blackboard.com/. Viewed 03 May, 2000.

[30] WebCT. Website. Available http://www.webct.com/. Viewed 03 May, 2000.

[31] IMS Global Learning Consortium, Inc. IMS Enterprise Information Model Version 1.01. Last revised December 21, 1999. Available http://www.imsproject.org/enterprise/eninfo03.html. Viewed 03 May, 2000.

[32] IMS Global Learning Consortium, Inc. IMS Question & Test Interoperability Information Model Specification Version 1.0 - Public Draft Specification. 18 February, 2000. Available http://www.imsproject.org/question/qtinfo01.html. Viewed 03 May, 2000.

[33] Advanced Distributed Learning Initiative. Sharable Courseware Object Reference Model (SCORM) Version 1.0. January 31, 2000. Available http://www.adlnet.org/Scorm/downloads.cfm. Viewed 03 May, 2000.

[34] *Ibid*., p. 28.

[35] World Wide Web Consortium. Extensible Markup Language (XML) . Document created April, 1997, last revised May 3, 2000. Available http://www.w3.org/XML/. Viewed May 3, 2000.

[36] The Bible. Genesis, Chapter 1, verse 1. Available http://www.genesis.net.au/~bible/kjv/genesis/. Viewed 03 May, 2000.

[37] Advanced Distributed Learning Initiative. Sharable Courseware Object Reference Model (SCORM) Version 1.0. January 31, 2000. Available http://www.adlnet.org/Scorm/downloads.cfm. Viewed 03 May, 2000. Page 32.

[38] World Wide Web Consortium. XML Schema Part 0: Primer. W3C Working Draft, 07 April, 2000. Available http://www.w3.org/TR/xmlschema-0/. Viewed 03 May, 2000. Schemas are more recent implementations of the (perhaps) more familiar DTD (Document Type Definitions).

[39] Dublin Core Metadata Initiative. Website. Available http://purl.org/dc/. Viewed 03 may, 2000.

[40] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1: Reference Description. 2000. Available http://purl.org/dc/documents/rec-dces-19990702.htm. Viewed 03 May, 2000.

[41] Microsoft. FrontPage. Website. Available http://www.microsoft.com/frontpage/. Viewed 03 May, 2000.

[42] Macromedia. Dreamweaver. Web site. Available http://www.macromedia.com/software/dreamweaver/. Viewed 03 May, 2000.

[43] For more information, see the WAP portal: WapCom. Website. http://www.wap.com/. Viewed 04 May, 2000.

[44] World Wide Web Consortium. Extensible Stylesheet Language (XSL). 1997. Last modified 04 May, 2000. Available http://www.w3.org/Style/XSL/. Viewed 04 May, 2000.

[45] Paille, G., Norman, S., Klassen, P. and Maxwell, J. The Effect Of Using Structured Documents (SGML) In Instructional Design. NAWeb Conference Proceedings, October, 1999. Available http://www.unb.ca/wwwdev/naweb99/proceedings/paille. Viewed 04 May, 2000.

[46] For example, see: Open Learning Agency. Open School Courses and Resources Outlines. 1999. Last modified, 21 February, 2000. Available http://www.openschool.bc.ca/outlines/. Viewed 04 May, 2000.

[47] For example, Auto-Graphics' Smart Editorial System (SES). Website. Available http://www.auto-graphics.com/pubtools.html. Viewed 04 May, 2000.

[48] IMS Global Learning Consortium, Inc. Specifications. 2000. Web site. Available http://www.imsproject.org/specifications.html. Viewed 04 May, 2000.

[49] World Wide Web Consortium. W3C's Math Home Page. Web Site. 1998. Last revised, April 28, 2000. Available http://www.w3.org/Math/. Viewed 04 May, 2000.

[50] World Wide Web Consortium. Synchronized Multimedia. Web Site. 1998. Last modified, May 3, 2000. Available http://www.w3.org/AudioVideo/. Viewed 04 May, 2000.

[51] Half-Baked Software. Hot Potatoes. Web site. Available http://web.uvic.ca/hrd/halfbaked/. Viewed 04 May, 2000.

[52] Such as LessonBuilder, by INNOVA Multimedia Ltd. Website. Available http://www.innovamultimedia.com/lbuilder.htm. Viewed 04 May, 2000.

[53] Brickley, Dan. TML Language Specification. Institute for Learning and Research Technology, University of Bristol. Undated. Available http://www.ilrt.bris.ac.uk/netquest/about/lang/. Viewed 08 May, 2000.

[54] Brickley, Dan. Tutorial Markup Language (TML). Institute for Learning and Research Technology, University of Bristol. Undated. Available http://www.ilrt.bris.ac.uk/netquest/liveserver/TML_INSTALL/doc/tml_user.html., Viewed 08 May, 2000.

[55] Brickley, Dan. Public domain TML software. Institute for Learning and Research Technology, University of Bristol. Undated. Available http://www.ilrt.bris.ac.uk/netquest/about/soft/. Viewed 08 May, 2000.

[56] Crisp, Joel, and May, Paul. This is an example tutorial for Chemistry. Web site. Undated. Available http://www.ilrt.bris.ac.uk/netquest/liveserver/cgi-bin/tml.pl/netquest/liveserver/qbanks/demos/paul/chemtute/chemtute.tmlo?Intro=/netquest/about/demos/. Viewed 08 May 2000.

[57] Alpha Centauri's Universe. Deepest Impacts: A Species Demise E.L.E. Website. Available http://www.to-scorpio.com/link3e.htm. Viewed 04 May, 2000.

[58] DeLong, J. Bradford. Multimedia Page. Website. March or October, 1998. Available http://econ161.berkeley.edu/multimedia/Multimedia.html. Viewed 04 May, 2000.

[59] As a designer for a Grade 12 mathematics course in Manitoba, I had to deal with the fact that some specific calculator software was actually embedded into the western protocols mathematics curriculum. See Western Canadian Protocol, The Common Curriculum Framework for 10-12 Mathematics. June, 1996, p. 99. Available http://www.wcp.ca/math/10-12/cluster/applied.pdf. Viewed 04 May, 2000.

[60] Seymour, Jim. How Application Service Providers Will Change Your Life. *The Street.Com* June 28, 1999. Available http://www.wsaccess.com/theStreet/comment/techsavvy/759956.html. Viewed 04 May, 2000.

[61] Clark, Ruth. Seminars (outlines only). Website. Available http://www.clarktraining.com/seminars.html. Viewed 09 May, 2000.

[62] Wieseler, Wayne. RIO: A Standards-Based Approach for Reusable Information Objects. Cisco Systems White Paper. Undated, p. 9. Available http://www.coursenet.com/html/products/white_paper_RIO.htm in pdf format. Viewed 09 May, 2000.

[63] Wieseler, Wayne. RIO: A Standards-Based Approach for Reusable Information Objects. Cisco Systems White Paper. Undated, pp. 9-10. Available http://www.coursenet.com/html/products/white_paper_RIO.htm in pdf format. Viewed 09 May, 2000.

[64] Ohio Schoolnet. Lesson Planning Template. Website. Available http://tlcf.osn.state.oh.us/blueprint/index.html. Undated. Viewed 09 May, 2000.

[65] IMS Global Learning Consortium, Inc. IMS Learning Resource Meta-data Best Practices and Implementation Guide Version 1.0. Web site. Available http://www.imsproject.org/metadata/mdbest01.html. Viewed 05 May, 2000.

[66] See, for example, Liljegren, Jonas. RDF Schema editor. Web site. Last updated 27 February, 2000. Available http://paranormal.se/perl/proj/rdf/schema_editor/. Viewed 08 May, 2000. Liljegren's schema editor retrieves schemas and, on the basis of the retrieved schema, generates a form for inputting new objects.

[67] Schema.net. Website. Available http://www.schema.net. Viewed 05 May, 2000.

[68] IMS Global Learning Consortium, Inc. IMS Content Packaging: Packaging Information Model Version .92 - Public Draft Specification. Web site. Available http://www.imsproject.org/content/cpinfo02.html. Viewed 05 May, 2000.

[69] Microsoft. Copnceptual Model. Microsoft Learning Resource Interchange Format. Entire LRN available http://download.microsoft.com/download/win2000srv/Install/1.00.0210.00/WIN98/EN-US/LRNSetup.EXE and the file will be found at Microsoft LRN Toolkit/Spec Samples/LRNSpecOnePackage/conceptual.htm. Viewed 05 May, 2000.

[70] Martindale's Health Science Guide. The "Virtual" Medical Center. Web Site. Available http://www-sci.lib.uci.edu/HSG/Medical.html. Viewed 04 May, 2000.

[71] I-Syndicate. Web Site. Available http://www.isyndicate.com/. Viewed 23 May, 2000.

[72] Individual.Com. Web Site. Available http://www.individual.com/. Viewed 23 May, 2000.

[73] Netscape. NetCenter. Web Site. Available http://my.netscape.com/. Viewed 23 May, 2000.

[74] Netscape. Create Your Own My Netscape Channel. Available http://my.netscape.com/publish/. Viewed 23 May, 2000.

[75] Barron, Tom. A Portrait of Learning Portals. ASTD Learning Circuits, May, 2000. Available http://www.learningcircuits.com/may2000/barron.html. Viewed 23 May, 2000. See also Karrer, Anthony. Building a Learning Portal. ASTD Learning Circuits, May, 2000. Available http://www.learningcircuits.com/may2000/may2000_ttools.html. Viewed 23 May, 2000.

[76] TraininkTek.Com. Web Site. Available http://trainingtek.com/. Viewed 23 May, 2000.

[77] AOL-at-School. Web Site. Available http://www.school.aol.com/teachers/index.adp. Viewed 23 may, 2000.

[78] World Wide Web Consortium. Cascading Style Sheets. Website. Last updated April 25, 2000. Available http://www.w3.org/Style/CSS/. Viewed 08 May, 2000.

[79] Diagram adapted from Bourda, Yolaine and Hélier, Marc. What Metadata and XML Can Do for Learning Objects. Webnet Journal, Volume 2, Number 1, January-March, 2000, p. 29.

[80] Suzuki, Junichi and Yamamoto, Yoshikazu. Building A Next-Generation Infrastructure for Agent-based Distance Learning. 2000. Forthcoming: International Journal of Continuing Engineering Education and Life-Long Learning. Available http://www.yy.cs.keio.ac.jp/~suzuki/project/pub/ijceell.pdf.zip (zipped PDF format). Viewed 08 May, 2000.

[81] Baucus, David. e-learning embedded in corporate work processes. Post to corped (mailing list), May 5, 2000. Available http://www.egroups.com/message/corped/1765?&start=1739. Viewed 08 May, 2000.

[82] My thanks to Terry Butler for this phrasing and for motivating the polemic which follows.

Learning object definitions vary greatly, however the basic idea behind learning objects are as follows: â€œThe smallest independent structural experience that contains an objective, a learning activity and an assessment.â€ (Lâ€™Allier, 1997). Learning objects are essentially like pre-developed learning materials that are self-contained and can easily be used in your eLearning programs. These objects can also be aggregated to produce lengthier learning experiences. The goal is to use LOâ€™s in your course in order to minimize the time needed to complete course development. Itâ€™s All in the Metadata.