Canadian Journal of Learning and Technology
Volume 28(3) Fall / automne, 2002

Great Idea, but how do I do it? A practical example of learning object creation using SGML/XML

Janet Bartz

About the Author

Janet Bartz is an Instructional Media Architect for the Open Learning Agency, working with the SGML/XML content development model. Address: 1117 Wharf Street, Victoria, BC, phone: 250 953 7422, email: janetb@ola.bc.ca

Abstract

The educational community is interested in learning objects, what they are, how they are used, and the many benefits derived from their use. Most educators are familiar with the value of learning objects in theory, but on the practical side are wondering what is involved in creating them. This paper offers a "how-to" of learning object implementation, for text, based on four years experience working with Open Learning Agency’s structured content development model. Throughout the paper the analogy of a yogurt container will be used to help illustrate the concepts behind implementing a structured content development model.

Rsum

La communauté des éducateurs est intéressée par les objets d’apprentissage, par ce qu’elles sont, par la façon dont on les utilise et par les nombreux avantages qui découlent de leur utilisation. La plupart des éducateurs sont familiarisés avec la valeur des objets d’apprentissage en théorie, mais du point de vue pratique, ils se demandent ce qui participe à leur création. Cet article propose une marche à suivre de la mise en place des objets d’apprentissage, pour le texte, fondée sur une expérience de travail de quatre ans avec le modèle de développement de contenu structuré de l’Open Learning Agency (Agence d’éducation ouverte). Nous avons utilisé, tout au long de l’article, une analogie avec un pot de yogourt pour tenter d’illustrer les concepts qui sous-tendent la mise en place d’un modèle structuré de développement axé sur le contexte.

Introduction

A learning object is any piece of information that can be used to contribute to a learning experience. The IEEE Learning Technology Standards Committee defines learning objects as "…any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning." (2001). Learning objects can take many formats including instructional videos, websites, textbooks, illustrations from textbooks, or audio clips. They can be any size; for example a full course and a single graphic could both be described as learning objects. The format and size of a learning object depends on the need the object was created to meet, including its potential for reuse. The value of learning objects is in their reuse. The Open Learning Agency (OLA) is a publicly funded BC learning establishment involved in the development and delivery of distributed learning materials for university, college, K-12, and workplace training. OLA has been using a structured content development model for creation of its text-based learning objects since 1998 and in the fall of 1999 published fourteen senior secondary courses that were developed using this model. Implementation of a structured content development model was undertaken as course content was required to be output in both print and online media. The efficiencies of developing and maintaining one instance of content, in a platform and software independent format, is an important feature as well as the potential for re-use. For example, content developed for Information Technology grade 11 and 12 courses is output as two secondary courses (

http://k12online.ola.bc.ca/index.html

) and is re-purposed for use in The Learning Lab (

http://www.ola.bc.ca/tll

), a teacher's professional development program (Porter, 2001).

Structured Content Development Model

To implement a structured content development model, the underlying structure for all text learning objects must be designed. The learning objects themselves are authored in accordance to this defined structure using a markup language such as SGML (Standard Generalized Markup Language) or XML (eXtensible Markup Language). Authored content in the form of SGML/XML files resides in a content repository, which has capabilities for search, retrieval, revision, and version control, among other things. Acquisition and administration of such a repository is required. For output, the SGML/XML file must be transformed for a specified media. The transformation process requires a programmer to develop custom scripts, a technician to run the scripts once developed, and visual designers to create the look of the output media such as web pages, print products, and CD-ROMs.

Developing educational content using a structured content development model requires a great deal of effort. For organizations that create a lot of similar content, the effort is worthwhile as content can be output in a variety of media, with a consistent look within each output media. Also, the concept of content that can be reused in different contexts, both within and outside an institution, is the underlying theme behind the "economics" of learning objects (Downes, 2000).

Tools for Structure — Markup Languages

The "structure" in a structured content development model is obtained through the use of a markup language. OLA's model was originally implemented using SGML but is currently being transposed into XML. Using a markup language allows one to define structure and identify all content authored to that structure by what it is, not what it looks like (Arbortext, 1995). SGML was developed in the early 1970s becoming an ISO standard in 1986 (Goldfarb & Prescod, 1998). XML is a subset of SGML, developed for the Internet and released as a World Wide Web Consortium specification in 1998 (W3C, 1998). A markup language allows one to define structure for a document and populate the document with content where content is identified by what it is (semantics), not by what it looks like (style). HTML (Hypertext Markup Language) was also developed for the Internet, however, it is based purely on the presentation of content (style) (Goldfarb & Prescod, 1998). With the evolution of desktop publishing, there has been much emphasis on the look of a document. Writers not only write content, but also spend hours formatting so it has a pleasing look. More sophisticated users set up style sheets in their word processing or desktop publishing software in order to apply consistent styles throughout their documents. Usually these styles have names like body text, heading, and subhead . This naming reflects the style applied to the content, not what the content is. For example, body list may be applied to a list of learning outcomes and a list of required resources for a lesson. It may be visually attractive to style these two items in the same way, however, semantically they are very different. Using a markup language to define the content, the learning outcomes may be identified with a tag called learningoutcomes and the resources with a tag called resources. In output to the specified medium, the content in learningoutcomes and resources could have the same style applied to them, or different styles. The power of identifying content for what it is lies in the ability to intelligently search content for reuse. For example, an instructor able to search for learningoutcomes can easily and efficiently find and evaluate existing content. In the same way, if a certain textbook is already being used, searching for it in resources is an efficient way to find relevant content. It is advantageous to use a markup language for educational content because: A markup language allows you to define content for what it is, thereby creating a database of content that can be easily searched and reused. Defining a structure ensures consistency and completeness in all documents of that type. Content is independent of format and style (these are applied at output when the medium is specified). Corrections only need to be done in one file, not in every output instance of the document. For example, if a typo is found in a course delivered both in print (developed in desktop publishing software) and online (developed in HTML), the correction must be made in two places. If the course were developed using a markup language, the correction would only need to be made in one place and the two output instances run again. If more extensive changes were required, the benefits of "single source" are even greater. SGML and XML are ASCII files and so are platform and software independent.

Designing the Structure — Document Type Definitions

The first step in developing a structured content development model is to define the document’s structure for the markup language. This structure is referred to as a Document Type Definition (DTD). A DTD is a "rule set" (Maler & El Andaloussi, 1996, p.4) to identify what pieces of content (defined as "elements") are required or allowed at what place in the document. Elements such as learningoutcomes and resources are created, and rules such as "learningoutcomes are required in a lesson, but resources are optional" (there may be no resources required for a lesson) are defined. Once the structure is defined, content is written to the structure using a markup language (SGML or XML). The DTD is a text file that is referenced by the conforming SGML or XML file that holds the authored content. To create a valid SGML or XML file, a parser is used to compare the file to the DTD to ensure it conforms to the structure defined in the DTD.

Creating a DTD involves a detailed analysis of the current set of documents it needs to represent (Maler & El Andaloussi, 1996). As part of the analysis, group discussions with individuals involved with the document creation process need to be conducted. This includes working with instructors, course designers, writers, editors, desktop publishers, and end users of the document to find out what is important, what works well, and where problem areas are. In analysis, look for patterns and anomalies and clarify if what breaks a pattern should be allowed or not. This structure will be the defining blueprint from which content is authored. An SGML or XML file is not valid unless it conforms to the defined DTD so it is important to involve all stakeholders and be sure they understand the DTD will define the standard for the organization’s document creation. Group consensus is needed in decisions regarding the structure of the document.

As an example, the text for a reading journal activity follows. In analysing the meaning of the content of the activity, three distinct pieces are identified: title, instructions, and criteria. In looking at other activities, one would compare if these three pieces were common to all activities and if other pieces were needed such as introduction, associated marks, exemplar, etc. In studying a group of activities, it might also become apparent that one activity type was not sufficient, and that activities should be modelled depending on what type of activity they were. For example, a reflection activity such as the reading journal is a much simpler structure than an activity with set question types such as multiple choice.

title instructions	Reading Journal It is useful to keep a journal to record your impressions, thoughts, questions, and conclusions about what you’ve read. Writing helps you to understand what you’re reading as you can work through your thoughts by writing them down. Writing can also help you to learn to read more carefully and with greater attention to detail. Notes you make while reading will also assist you later when you write your essay in the section assignment. As you work your way through the lessons in this section, you’ll be reading many short stories from your textbook. After reading a story, you should record your response in a reading journal. It may also be helpful to stop and make notes in the journal as you are reading the story. To set up a reading journal, go to the back of the workbook to the "Journal" pages. Enter the appropriate information in each column. Under the "Response" heading, enter your personal impressions which may include things such as initial observations and questions while reading, reflective impressions, and how the writing style and elements of fiction contribute to the work’s meaning.
criteria	A successful response entry will include thoughtful, personal comments on your reactions and impressions of the work — not simply a summary of key events without reflection.

Another factor to keep in mind when designing a DTD, is to design structures to hold the type of information that would accompany authored content, but is not output for viewing. For example, it might be useful to include information such as author and date of creation for the reading journal activity above. Information such as this is referred to as metadata.

One of the most difficult decisions in designing a DTD is establishing how detailed to make it. A good representation of structure with enough rigidity to force compliance and with enough flexibility to handle all the different nuances within documents, is desired — detailed enough to handle all possibilities yet simple enough to be easy to use. Too detailed a structure is difficult to write content to and more difficult to process for output, as the more elements defined, the more must be written to and addressed. A loose structure results in allowing elements to be interpreted and used in different ways by different authors. Knowing the authors and their tolerance for following a set structure can be used as a guide to define how rigid or loose to make the model. If it is too complex, people will not want to use it.

Development of a DTD is an iterative process. Once defined and implemented, users will request changes based on things they need that were not included or things that are not working well for them. The best way to see if the structure will work is to try it out with actual content as a beta project. There is a significant learning curve as users learn to develop within a structured environment. This is to be expected as change and establishment of new methods is a lengthy process. It is best to start with a simple structure and build on it as stakeholders and users become more experienced working within a structured environment and more knowledgeable about what it can and can not do. At OLA we are currently revising our original DTD based on feedback from four years' use.

A DTD is a defined, hierarchical structure that can be thought of as a collection of containers to hold and identify specified objects. Using an analogy, if you wanted to purchase yogurt you would go to a food store and look for the yogurt in the dairy case. The food store can be thought of as a structured environment, a container for food that can be purchased. The dairy case too is a structure, a container for dairy-based foods. The yogurt container is a structure designed to contain yogurt. The labelling on the container provides information about the kind of yogurt. Just as the yogurt is contained within a dairy case and food store, a learning object, such as an activity, is contained within a course structure.

When designing DTDs, diagrams are often used to visualize the structure before writing occurs. Following are two tree diagrams (Maler & El Andaloussi, 1996) of simplified structures. Elements are depicted as boxes. No occurrence indicator beside an element means that the element is required and can only occur once. The "?" means there can be zero or more elements, "+" means one or more elements are required, and "…" means that part of the diagram is described elsewhere. For example, in the case of the course structure under assessment there are three choices: reflection, practice, and assignment. There must be at least one assignment within assessment, and there can be none or more reflection and/or practice activities. In this diagram the structure for reflection is modelled, however, the structures for practice and assignment are not. To keep the examples clear, the structures represented are greatly simplified from how they would be designed for actual application.

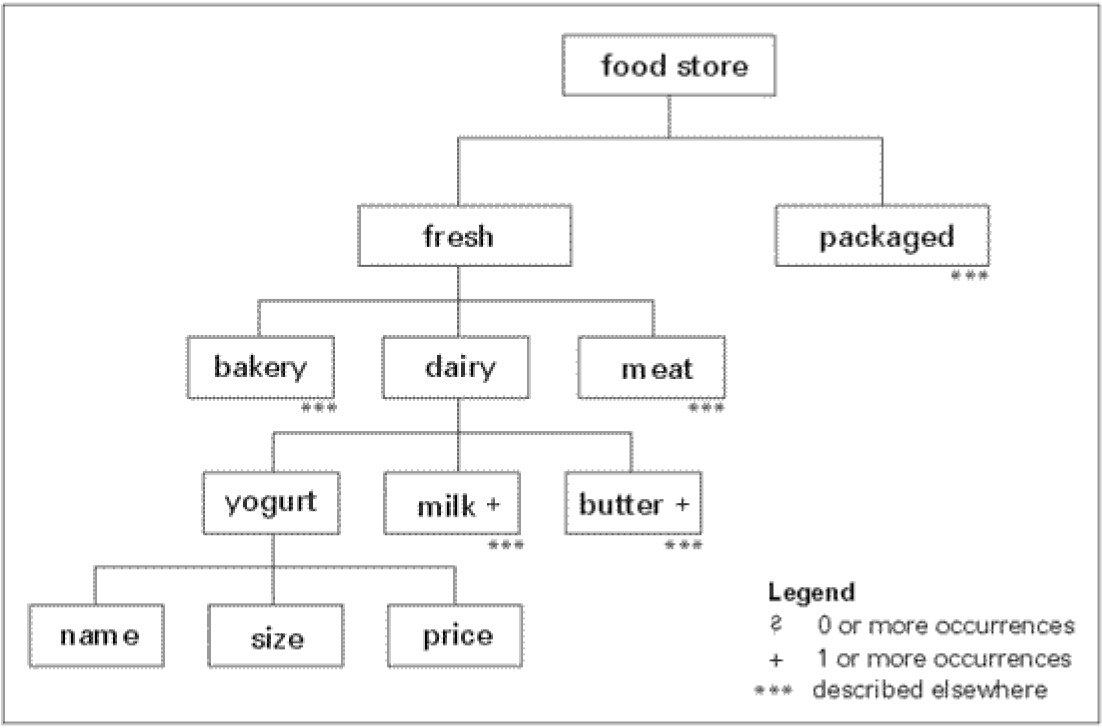


Figure1. Tree diagram showing simplified food store structure.

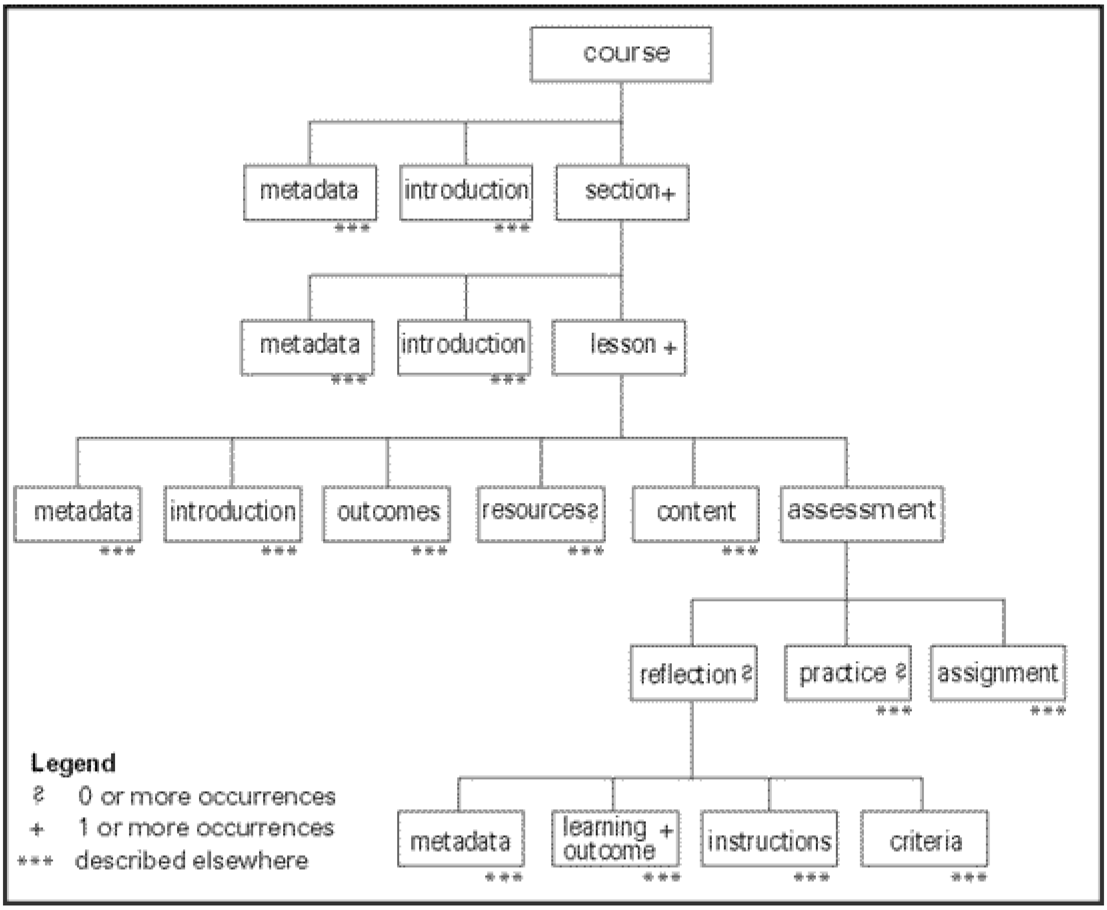
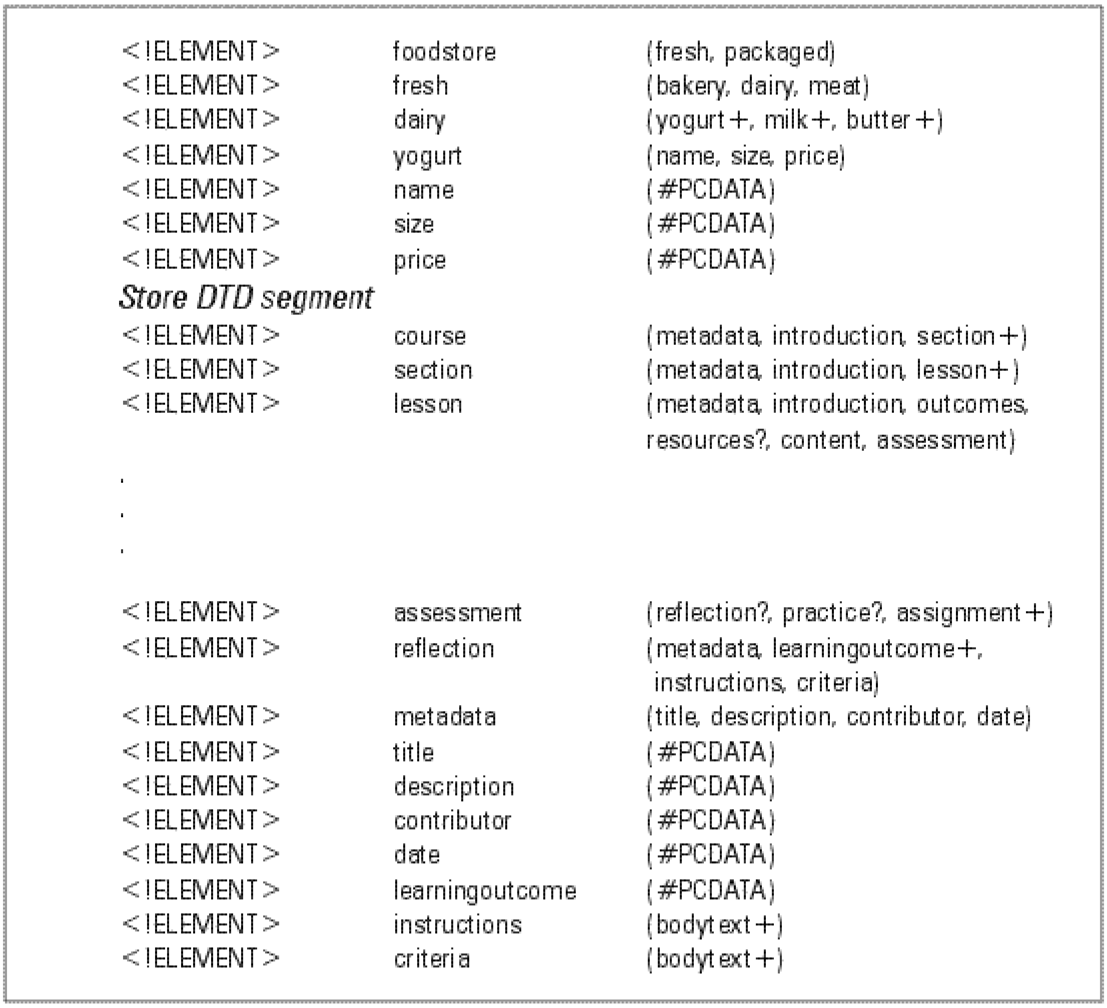


Figure 2. Tree diagram showing simplified course structure.

Figure 3 shows segments of DTDs written for the food store and course structures. DTDs define all elements and their relation to each other so complete DTDs would include definitions for all elements in the structure. ("PCDATA" stands for parsed-character data, which means it is text or "data characters" (Maler & El Andaloussi, 1996, p. 22).)

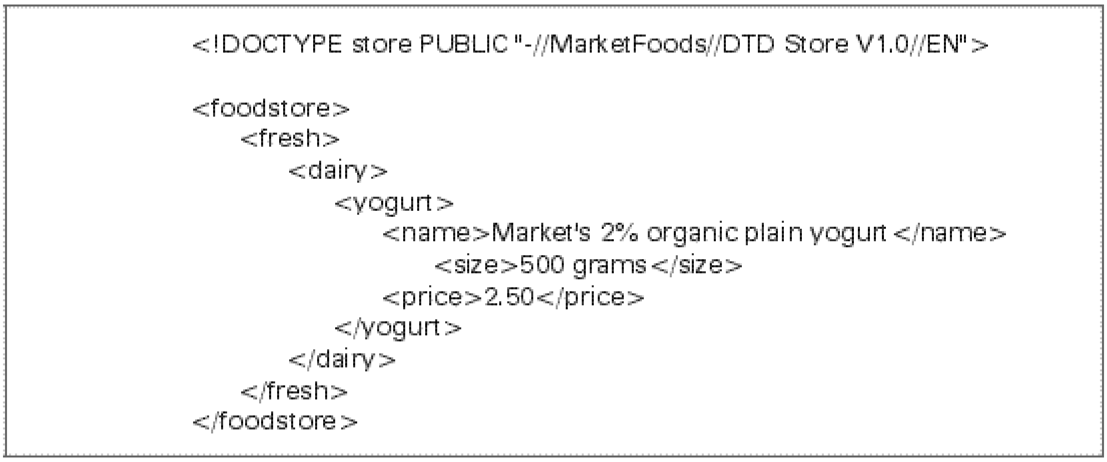


```
<!ELEMENT>        foodstore           (fresh, packaged)
<!ELEMENT>        fresh               (bakery, dairy, meat)
<!ELEMENT>        dairy               (yogurt+, milk+, butter+)
<!ELEMENT>        yogurt              (name, size, price)
<!ELEMENT>        name                (#PCDATA)
<!ELEMENT>        size                (#PCDATA)
<!ELEMENT>        price               (#PCDATA)
Store DTD segment
<!ELEMENT>        course              (metadata, introduction, section+)
<!ELEMENT>        section             (metadata, introduction, lesson+)
<!ELEMENT>        lesson              (metadata, introduction, outcomes,
                                      resources?, content, assessment)
.
.
.

<!ELEMENT>        assessment          (reflection?, practice?, assignment+)
<!ELEMENT>        reflection          (metadata, learningoutcome+,
                                       instructions, criteria)
<!ELEMENT>        metadata            (title, description, contributor, date)
<!ELEMENT>        title               (#PCDATA)
<!ELEMENT>        description         (#PCDATA)
<!ELEMENT>        contributor         (#PCDATA)
<!ELEMENT>        date                (#PCDATA)
<!ELEMENT>        learningoutcome     (#PCDATA)
<!ELEMENT>        instructions        (bodytext+)
<!ELEMENT>        criteria            (bodytext+)
```

Figure 3. Course DTD segment.of this example. To keep things simple, the example does not contain content for assignment, which is a required element within assessment.

Authoring to a Structure

Once the structure (DTD) is defined it can be filled with authored content. In our example of a yogurt container as the defined structure, the content is the yogurt. While any type of yogurt could be placed in the container, in this example it is Market’s 2% organic plain yogurt. This would be marked up as shown in Figure 4 (the first line indicates what DTD to associate this marked-up content with).



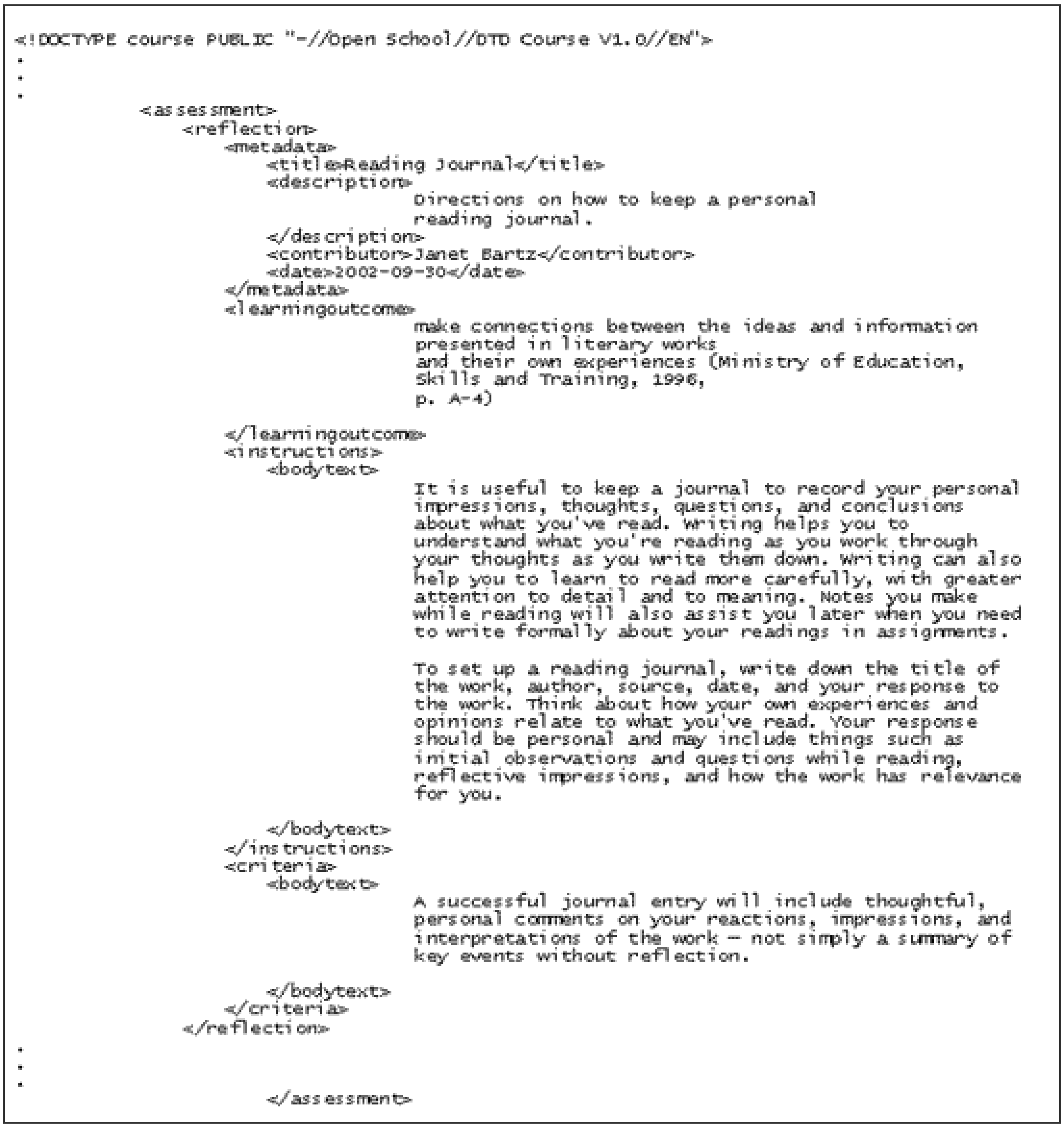
```
<!DOCTYPE store PUBLIC "-//MarketFoods//DTD Store V1.0//EN">

<foodstore>
    <fresh>
        <dairy>
            <yogurt>
                <name>Market's 2% organic plain yogurt </name>
                    <size>500 grams</size>
                <price>2.50</price>
            </yogurt>
        </dairy>
    </fresh>
</foodstore>
```

Figure 4.

Marked-up content for yogurt example.

Following, the reading journal activity is marked up using the assessment reflection structure as defined in the Course DTD. The original content was revised to make it a standalone learning object — references to other parts of the course (lessons and sections) and media specific references (workbook) were removed. Information abut the activity has also been added so the content now consists of metadata, learning outcome, instructions for the learner, and evaluation criteria. Some parts of the metadata, such as title, will be output for the learner to view, while others, such as date and description are only for indexing purposes. A more extensive set of metadata would be used to index a "real" activity, just as a "real" learning object and course would contain a structure much more complex than that. To keep things simple, the example does not contain content for assignment, which is a required element within assessment.

Part of writing a DTD is writing accompanying documentation including an element dictionary (to define all elements in the structure) and best practices guide (the best way to write content for the structure). Documentation such as this is necessary so that content is structured in a consistent and meaningful manner. For example, one could argue that the reading journal activity might be more appropriate in the content structure as it is not formally assessed. If, in the best practices guide, there are directions to put any writing or other creation action of the learner within the assessment structure, and any activity to do with personal opinion in the reflection structure, it is clear where the reading journal activity should go.



```
<!DOCTYPE course PUBLIC "-//Open School//DTD Course V1.0//EN">
.
.
.
            <assessment>
                <reflection>
                    <metadata>
                        <title>Reading Journal</title>
                        <description>
                                      Directions on how to keep a personal
                                      reading journal.
                        </description>
                        <contributor>Janet Bartz</contributor>
                        <date>2002-09-30</date>
                    </metadata>
                    <learningoutcome>
                                      make connections between the ideas and information
                                      presented in literary works
                                      and their own experiences (Ministry of Education,
                                      Skills and Training, 1996,
                                      p. A-4)

                    </learningoutcome>
                    <instructions>
                        <bodytext>
                                      It is useful to keep a journal to record your personal
                                      impressions, thoughts, questions, and conclusions
                                      about what you've read. Writing helps you to
                                      understand what you're reading as you work through
                                      your thoughts as you write them down. Writing can also
                                      help you to learn to read more carefully, with greater
                                      attention to detail and to meaning. Notes you make
                                      while reading will also assist you later when you need
                                      to write formally about your readings in assignments.

                                      To set up a reading journal, write down the title of
                                      the work, author, source, date, and your response to
                                      the work. Think about how your own experiences and
                                      opinions relate to what you've read. Your response
                                      should be personal and may include things such as
                                      initial observations and questions while reading,
                                      reflective impressions, and how the work has relevance
                                      for you.

                        </bodytext>
                    </instructions>
                    <criteria>
                        <bodytext>
                                      A successful journal entry will include thoughtful,
                                      personal comments on your reactions, impressions, and
                                      interpretations of the work - not simply a summary of
                                      key events without reflection.

                        </bodytext>
                    </criteria>
                </reflection>
.
.
.
                        </assessment>
```

Figure 5. Marked-up content for reflection activity example.

Metadata — a Critical Part of the Structure

Metadata is "structured data which describes the characteristics of a resource" (Taylor, 1999). It is the information about the learning object that identifies it. For example, labelling makes it easy to identify that 2% organic plain yogurt is inside the container without having to open up the container and look inside. The type of yogurt (content) in the yogurt container (structure) is identified by the labelling (metadata). In the Store DTD example,

name, size,

and

price

are the elements that hold the metadata about the yogurt. In the Course DTD, the metadata elements (contained in the parent element

metadata

) are

title, description, contributor,

and

date

. All are required. In a structured information model, the metadata is part of the model, part of the structure (DTD) to be filled with content to make a valid file. One can think of metadata as an information piece that is attached to the core content. In the Course DTD there is metadata attached to course, section, and lesson. Although not shown on the tree diagram, there would also be metadata elements for each type of assessment and content. Thus each piece with accompanying metadata can be thought of as an identifiable learning object. A robust metadata set would contain information pertaining to areas such as object lifecycle, technical requirements, educational specifications, copyright, and classification (Fisher et al., 2001). When looking for learning objects in a repository, it is the information contained in the metadata that is searched. Therefore consistency in specification and application of metadata, across an organization or community, facilitates searching.

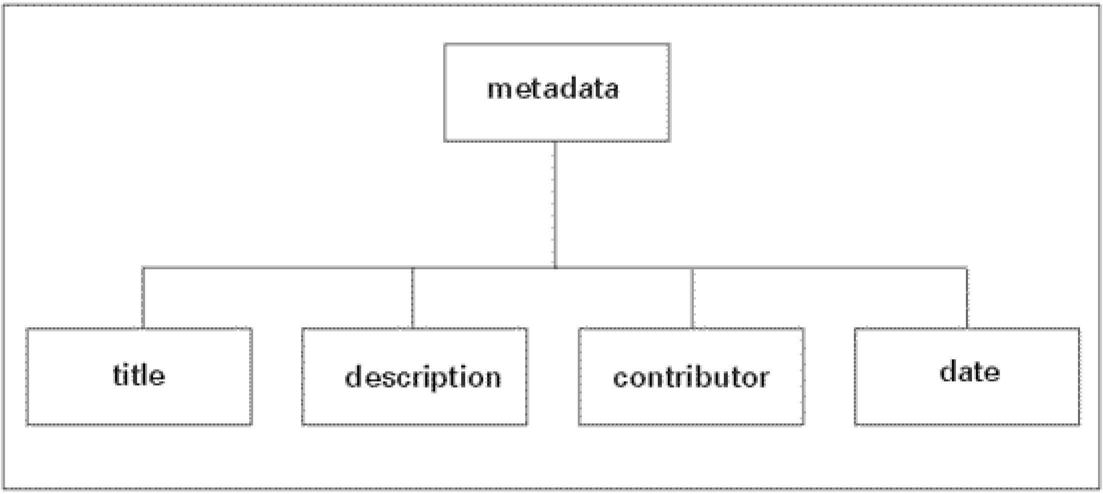


Figure 6.

Course metadata elements.

The concept of metadata for identifying resources on the Internet began in the mid 1990’s and was formalized with a meeting in Dublin, Ohio that was the beginning of the Dublin Core Metadata Initiative (DCMI, 2000). In the mid to late 1990s, consortiums began to form with an interest in the interoperability of learning objects such as: IMS (Instructional Management Systems,

http://www.imsproject.org/

), ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe,

http://ariadne.unil.ch/Metadata

), EdNA (Education Network Australia,

http://standards.edna.edu.au/metadata/

), and GEM (Gateway to Educational Materials,

http://www.geminfo.org/

). The work of Dublin Core was used as a base for these consortia, but was expanded upon to include educational metadata.

In 1998 IMS and ARIADNE submitted a joint specification to the IEEE from which the IEEE LTSC Learning Object Metadata (LOM) standard was built (IMS Global Learning Consortium, 2001). CanCore, a Canadian initiative has developed a metadata application profile based on the IMS metadata specification (http://www.cancore.ca).

In establishing the set of metadata to use for learning objects, one needs to keep in mind whether conformance to a standard or specification is important (how broadly will the learning object be shared?). Decisions about which elements of the standard or specification to use (most elements are optional) also needs to be made. One has to establish a balance between the amount of useful information to hold against the time and effort to enter and maintain the information. The possible exchange of learning objects within a community of organizations will also impact the choice of metadata.

Content Repository

A repository is a large storage area for objects that enables users to find and re-purpose learning objects (CANARIE, 1999; Hatala & Nesbit, 2001). In the case of our yogurt container, the repository would be a refrigerator. The refrigerator holds many objects of different types that can be easily found (hopefully). How well organized the refrigerator is results in how easy it is to find what is inside. We find what is inside based on the labelling of the containers (the metadata). For example, in the case of the reading journal activity, a search could be done on the title element using the keyword "journal". The description element could then be viewed to assess whether or not to investigate the content. A content repository is a database designed to hold structured documents. The repository includes features such as search, edit, access control, version tracking, reuse of elements within other documents through reference, and dynamic delivery to the web. In choosing a repository, consider the integration requirements of

systems, such as the learning management system, student registration and administration system, test bank, and external learning institution systems. Once the content repository is populated with learning objects, clearly identified by associated metadata, there exists a bank of information to draw upon. This bank may be only within an institution or may be part of a broader community of institutions.

Output

We take the yogurt (content) out of the refrigerator (repository) to use in a variety of ways, such as eating it plain, mixing it with other ingredients to make a salad, or stirring it into a curry. We can do this because we know we have plain yogurt by the labelling on its container (metadata) and plain yogurt can be reused in many ways. Output of learning objects is similar in that they can easily be found (due to their metadata) and reused (due to how they are written). For example, the reading journal activity could be used for a number of English or literature lessons for recording personal impressions of short stories, poetry, or novels. It can be reused because it is written in a general manner, not referencing any specific literary works, course structures, or media.

Learning objects reside as SGML/XML files in the repository and can be output to a variety of mediums. This is possible because all or part of the file structure can be referenced and the content inside is format and media independent. To output, transformation scripts need to be written to act on the file and extract the required pieces, order them, and save them in an appropriate file type for output. The script also associates the visual styles for the output file. For OLA secondary courses, there is one set of transformation scripts to create HTML files for online output, and another script that creates an SGML file which is opened in software and converted to PostScript for print output.

A Collective Approach to Structure

A structured content development model is a collective model. A group of interested stakeholders defines the structure for the documents. The group also decides on other issues so the learning objects they write can be interchanged. This means agreeing on the writing style, the level of granularity, the audience

being written to, and possible output media. All members have to agree and adhere to group decisions — this is often a difficult thing to achieve. People have ownership over the materials they have developed and feel they are unique. There is a reluctance to work in accordance with a defined structure and one often hears the complaint about "hindered creativity." In practice, developers already write to a structure, for example, a course consists of lessons and lessons consist of topic exploration and practice activities. The structured information model formalizes that structure and if well designed, can handle all nuances required of it. People have been writing creatively to defined structures for centuries, for example, poetic structures such as the couplet, sonnet, or haiku.

Issues in Writing to a Structure

Style Issues

In writing content for learning objects, there are several important things to keep in mind. The first is that the defined structure of the DTD must be followed if the objects are being developed using a markup language. Even if a structured content development model is not in place, writing content as learning objects still requires certain considerations such as being standalone — granular, format and media independent, and broken into discreet, meaningful chunks.

Standalone content is content that is self-contained. Self-containment facilitates use and reuse of learning objects as they may be combined in any number and sequence. To be self-contained, a learning object should not make explicit reference to objects outside itself. For example, the revised reading journal activity does not make reference to a workbook or a specific story or poem. Directions to read particular works are separate from the activity, so that the activity is standalone making its potential for reuse much higher.

Granularity refers to the size of the component that defines a self-contained learning object or "the degree of precision with which learning objects … can be described" (Porter, 2001, p. 49). How granular the object is should be decided upon before writing so the writer knows the level of self-containment to write to.

It could be as large as a course or as small as an activity within a lesson. The smaller the piece, the more potential for reuse, but if too small can be difficult to write to. Also, contextualizing many small pieces all the pieces in a meaningful way can be challenging.

Writing content that is separate from format and media is also important. This involves avoiding specifics such as "glossary terms will be in shown in blue" or "bold face" as this may not be the case in all output media. Language must be as non-specific as possible in terms of format and media. Write to the meaning, not the look or location of the content. Statements such as "look at the green area on the map on the next page" should be written as "look at the Black Land area beside the Nile River on the following map of Egypt". Once a writer becomes familiar with this style of writing, it is not difficult to write standalone content that still provides clear instruction.

As content is self-contained, it is necessary to break it up into distinct chunks where each chunk contains only information relevant to itself. This set of information should be given a meaningful title. Keeping this in mind while authoring contributes to clearer instructional writing as the writer strives to separate concepts into discrete pieces rather than mixing them (Kilian, 2000). For example, a lesson may be written on the importance of plot in a story and include a discussion of the plot in a sample story from a textbook. It would be more useful to write two separate chunks in this case, one on the importance of plot and another on the plot in the sample story. The possibilities for reuse of the object on plot in general expands as it is now independent of the sample story (and an external resource).

Production Issues

SGML/XML files are simple text files and can be created using any text editor. Using an authoring tool, however, makes writing to the structure much easier, especially if the DTD is complex. Authoring tools provide visual representation of the DTD and let users know what elements are available to insert in any given point in the document. Using an authoring tool instead of a text editor is the same as hand coding HTML or using an HTML editor.

Even though an authoring tool makes placing content into its correct structure easier, it is sometimes difficult to have writers work directly with it, especially if the DTD is complex. The role of the writer is to provide good content, written to an organization's specifications or defined structure, but with an emphasis on the quality of the content itself. Good content can always be adapted to fit a defined structure; although, the writer must understand at least the higher level structures of the DTD and write in accordance with them. Poor content, even if it is well structured, is still poor content (though a structured environment does help to enforce completeness). If writers are required to write to the structure over a long period of time, it may be beneficial for them to learn to work directly in the authoring tool. For short term writing contracts, the learning curve is not worth the efficiencies gained by having the writer learn all the complexities of the DTD and use the authoring tool.

At OLA writers are typically contracted so it is unreasonable to expect them to purchase and learn to use a structured authoring tool for a short-term contract. Our solution is to provide a guidelines document explaining the structured development model along with word processing templates which define the structure (in broad terms). The files from the writers are converted into markup by a team of production staff dedicated to "tagging" content. Having a team of dedicated staff results in consistent markup and frees the writer to concentrate on the content. Throughout the process, however, the writer must be guided to provide content that will fit the defined structure of the model.

Team Support and Education

In implementing a structured content development model, it is important to recognize the value of educating the development team in this new method of content creation. Ongoing documentation and education throughout all phases of the model's development, testing, and implementation is a large but key task in the ultimate success of the model (Maler & El Andaloussi, 1996).

Effort must be made to educate development staff on the advantages of a structured information model. Involving staff early in the process in the design of the DTD will help them feel they have contributed to the architecture of the model. When people feel they have contributed to the design process, they are more likely to accept and use the resulting product rather than feel it is forced upon them. Also, to get the best representation of what the structure must contain, it is important to know the opinions of all contributors to the document. An editor will have a different perspective than a writer or desktop publisher, but all perspectives shape the document and are important.

One of the most difficult things about creating a definitive structure for a document is that there is only one. Design decisions must represent the general consensus, and deviations from the consensus must be weighed. It is useful to document decisions as they are made as design rationale may be questioned at a later date. Once the structure is defined, agreed upon, modelled, tested, and well documented with regards to use, it must be followed by all to create valid, consistent documents.

It is important for everyone to understand the model's entire process — from DTD design through authoring to output — to appreciate how the model works and what is and is not possible. Everyone must accept the collaborative nature of the model and understand that requested changes will affect all content in the model so must be decided upon by group consensus. Ongoing education and support for authoring, applying mark-up, entering metadata, using the repository, and output is required for consistency within the learning objects and success of implementation.

First Steps Towards Implementation

Even if implementing a structured content development model is a long way away, there are still steps that can be done now that will make going to such a model easier in the future. These include writing standalone content and defining a metadata set to represent the content. To write learning content that is standalone, do not reference other parts of the course or the format or media it is being presented in. Often this is fairly straightforward, simply substituting meaning for format. For example, instead of "look at the green line on the graph on the next page" use "look at the line representing growth on the following graph." Identifying the format, media, and transitional statements in writing, and creating standalone ways to address them, becomes easier with practice. The most important step is becoming aware of them and adapting the writing style to address the meaning of the content. Experience working with metadata can be gained by creating a simple database about existing content, no matter what form the content is in. (People already track some metadata such as title, author and contributors, date, revision history, and copyright; they just don't think of it as metadata.) Select a metadata application profile,like CanCore, and choose the elements that are appropriate to identify the content. The content itself does not have to be part of the database; a reference to where it resides is enough. This way the idea of searching and reusing learning objects can begin to happen. As user sophistication about metadata grows, the metadata set can be added to. Much of the work to date has concentrated on defining the metadata around the learning object, not on the structure of the object itself. For experience in using markup languages, a DTD could be written for the metadata set. Once the DTD is written, the metadata for the content can be structured in a markup language like XML.

Conclusion

As more institutions become involved in e-learning, the wealth of available learning objects will grow and the importance of finding an efficient way to handle the interoperability — the identification and exchange — of educational content will become a crucial issue. Using a markup language such as XML to define the metadata and the object itself is an obvious solution. In a structured content development model, information is represented by what it is, in a platform and software independent file type, and therefore can easily be found and reused. Even if an organization is not yet ready to undertake a structured content development model approach, writing standalone content and tracking metadata about the content are good first steps in preparing for participation in the learning object community.

Post Script

OLA has recently been awarded a contract by Industry Canada to act as the Technical Liaison Office (TLO) for EduSpecs. EduSpecs represents Canadian interests in the development of international e-learning standards and specifications. Their website is located at: http://eduspecs.ic.gc.ca/. The TLO will act as the lead in this initiative by:

- Representing Industry Canada on the IMS Technical Board and in the overall IMS specifications development process and in other e-learning specifications and standards initiatives as required.

- Identifying priority areas for Canadian involvement in national and international e-learning specifications and standards activities to ensure a coordinated and effective role for Canada.

- Establishing a focal point for coordinating and disseminating information on educational standards and specifications.

- Promoting and encourage networking between key players in the field, both within Canada and internationally, to leverage insights and knowledge and add weight to Canadian priorities internationally.

Acknowledgement

Thank you to Tracey Leacock, Assistant Professor, SFU, for her thorough review, edit and valuable suggestions.

References

Arbortext (1995). SGML: Getting Started. White Paper. Retrieved January 1998 from http://www.arbortext.com/html/white_papers_webinars.html

CANARIE (2002). Index of funding learning/1999 backgrounders. Retreived May 30, 2002 from www.canarie.ca/funding/learning/1999backgrounders/

DCMI (2000). History of the Dublin Core Metadata Initiative. Retrieved May 27, 2002 from http://www.au.dublincore.org/about/history/

Downes, S. (2000).

Learning Objects

. Retrieved May 27, 2002 from

http://www.atl.ualberta.ca/downes/naweb/Learning_Objects.htm

Fisher, S., Friesen, N., & Roberts, T. (2001).

CanCore Element Set 1.0

. Retrieved March 8, 2002 from

http://www.cancore.org/elementset1.html

Goldfarb, C. F., & Prescod, P. (1998). XML Handbook. New Jersey: Prentice Hall.

Hatala, M. and Nesbit, J. (2001). An evolutionary approach to building a learning object repository. In Computers and Advanced Technology in Education: Proceedings of the 4th IASTED International Conference . Calgary:ACTA.

IEEE Learning Technology Standards Committee, Learning Object Metadata Working Group (2001). Scope & Purpose. Retrieved March 8, 2002, from http://ltsc.ieee.org/wg12/s_p.html

IMS Global Learning Consortium (2001). IMS Learning Resource Meta-data Best Practices and Implementation Guide Version 1.1 - Final Specification. Retrieved January 1998 from http://www.imsglobal.org/metadata/mdbestv1p1.html

Kilian, C. (2000).

Writing for the Web: Writer’s Edition

. Self Counsel Press.

Maler, E. & El Andaloussi, J. (1996). Developing SGML DTDs from Text to Model to Markup.NJ: Prentice Hall.

Ministry of Education, Skills and Training, (1996). English Language Arts 11 and 12 Integrated Resource Package 1996. Province of British Columbia.

Porter, D. (2001, July). Object Lessons for the Web: Implications for Instructional Development. The Changing Faces of Virtual Education, Chapter 4, Commonwealth of Learning. Retrieved May 27, 2002 from http://www.col.org/virtualed/

Taylor, C. (1999). An introduction to metadata. University of Queensland Library. Retreived May 30, 2002 from www.library.uq.edu.au/iad/ctmeta4.html

W3C (1998).

Extensible Markup Language (XML) 1.0, W3C Recommendation 10-February-1998

. Retrieved March 8, 2002, from

http://www.w3.org/TR/1998/REC-xml-19980210

ISSN: 1499-6685