

USING DOCBOOK TO AID IN THE CREATION OF LEARNING CONTENT

Martínez-Ortiz, I.¹, Moreno-Ger, P.¹, Sancho-Thomas, P.², and Fernández-Manjón, B.²

¹ Centro de Estudios Superiores Felipe II. Madrid. Spain

² Facultad de Informática, Universidad Complutense de Madrid. Madrid. Spain

Abstract: Creating web based educational (e-learning) applications is a complex process that poses several challenges. We are especially concerned with two aspects: the cost of production and maintenance of educational contents, and the personalization of contents to suit individual learners. The Learning Objects (LO) model was proposed in response to reusability and adaptability concerns. IMS, ADL, IEEE and other initiatives have proposed standards for metadata description and packaging facilities in order to facilitate the interchange of LOs across Learning Management Systems (LMS). Metadata and packaging description demand additional work from content developers. In this paper, we address how DocBook, an XML/SGML document model, facilitates the creation of learning content and allows the countless DocBook materials already existing on the Internet to be reused. The main goals are to simplify the creation of adaptive learning content following the LO paradigm and to study how DocBook and its associated tools facilitate the authors' task when dealing with the XML documents needed for LO description and interchange.

Keywords: IMS, DocBook, Learning Object, XML.

1. INTRODUCTION

A teacher's primary task for preparing a course involves creating the learning content that will be given and covering the learning objectives of the course. This task is time consuming because teachers need to create or adapt preexisting content to fit target learners' knowledge. The material developed is not usually self-contained so the teacher must clarify complex parts in more detail or explain the key points of the lesson.

This complexity grows exponentially when developing educational content for an LMS since content that can be used by a wide range of learners needs to be developed. The difficulties for authoring and maintaining high quality educational contents have led to a large investment in creating multidisciplinary groups of experts for these tasks. But either due to the evolution of technology or migration processes to different learning platforms, content that was made available in a specific format is often lost because it is too expensive to adapt it to the new environment.

Solutions for tackling content data portability have been used in the documental domain (Goldfarb, 1981), (Coombs, 1987) using descriptive markup languages, in particular the Standard Generalized Markup Language (SGML). Nowadays this technology has been superseded by the eXtensible Markup Language (XML) (W3C, 2004). XML uses tags to structure information and defers presentation information to a separate document: an XSL file (W3C, 2001). The main ideas behind this proposal are:

1. Authors need only concentrate on creating and structuring content, not presentation.
2. Content and processing/presentation methods must be separated.

This approach can be applied in the educational domain because the problems are very similar. In addition, using XML based learning content is a first step in order to create machine readable content that can be processed by software applications to adapt the content itself to learners' needs.

As people's learning needs have recently changed, now training programs and training systems should fill the gaps in the knowledge necessary to carry out their jobs. To ease this new profile of formation a new model of learning has been promoted. The Learning Object model addresses problems of reusability and adaptability of educational contents (Koper, 2003), (Downes, 2001).

This model is based on the definition of reusable pieces of content, usually called Learning Objects (LOs), which can then be assembled to form complete courses. Ideally, LOs should be easily reassembled or adapted to

suit the needs of different learner profiles. The success of this model depends on the proposal of standard specifications that define the processes in order to create, catalogue and document the content generated.

This standardization effort must deal with concepts such as the facilitation of the interchange of LOs (i.e interoperability) or the maintenance of metadata describing the possible uses of each specific unit. To sum up, a content developer that follows a LO paradigm must deal with content development and its packaging according to standards that promote its reusability among different systems.

By using XML technologies such as XSL Transformations (XSLT) (W3C, 2001b) on the pre-existing material, the course development process could be simplified because some of the tasks (i.e. packaging process) can be automated or at least made more systematic.

Also, XML content is more versatile than the usual HTML content for adaptation purposes. For instance, information could be adapted to a specific user by means of different transformations of the same content (e.g. filtering difficult concepts for novice users). In the long run, this approach would produce a more systematic and industrial process for creating, updating and adapting e-learning courses.

The paper is structured as follows: in section 2 we present an overview of the basic structure and principal features of DocBook. In section 3 we describe our notion of the content author / revision / publishing cycle and the improvements obtained by using DocBook. In section 4 we present how to create content ready to be deployed in a Learning Management System (LMS) compliant with IMS Content Packaging specification. Finally some conclusions and future work are depicted in section 5.

2. DOCBOOK

DocBook is a markup language (i.e. a document model) defined by an SGML or XML Document Type Definition (DTD). A DTD is a description of all the elements that are permitted in any document that you make using that DTD. To be precise, DocBook DTD defines a vocabulary particularly well suited to represent books and papers about computer hardware and software, but by no means limited to that use (Walsh, 1999). DocBook was originally created as an SGML dialect developed by O'Reilly and HaL Computer Systems in 1991 with the objective of facilitating the interchange of software manuals between companies, and it quickly became a *de-facto* standard (Dougherty, 1999).

A DocBook document annotates the semantics of the document but not its appearance. In other words, DocBook tags describe only the structure of the document. This focus on document structure contrasts with the focus of word processors, where this information is usually lost because there is more stress placed on presentational aspects (e.g. fonts, size, etc). The idea behind this is to capture the mental processes made by the content creator in order to structure, that is to say, how he/she wants to transmit the underlying information and not the final result (print-ready version) of the document.

How DocBook elements are rendered is arbitrary, given the conceptual meaning of the text. In fact, these decisions should be made by the graphical designers, not the authors. DocBook gives you the essential structure of a document without attempting to render elements in WYSIWYG fashion.

```

<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.2//EN">
<book xml:lang="en">
  <title>DocBook Example</title>
  <section>
    <title>First Section</title>
    <para>This is the first sentence of
      first section of this book</para>
    ...
  </section>
  ...
</book>

```

Figure 1. DocBook Document Sample

The DocBook document model provides support for information about main hierarchical structures (e.g. set, book, chapters, appendix, bibliography, etc.), hierarchical content (e.g. sections, glossary, etc.), multimedia content, references, questions and answers, admonitions and metadata.

In addition, DocBook provides support for specific features related more to software: Programming language features (function/methods, classes, etc.), Generic Input / Output user interaction, Graphical User Interface interaction, etc.

Apart from separating content and appearance, DocBook-style conceptual markup lets you work with element types systematically. For example, it may be of interest to make our learning content accessible for people with disabilities. W3C proposed a set of guidelines (W3C, 1999) that were intended for developers of web and authoring tools, but these

guidelines can also apply to our DocBook documents. As a simple example, we can generate a report that shows us the results when checking if our content contains alternative text descriptions for every piece of multimedia content.

DocBook is built in such a way that it can be used to author any kind of modular content. That is, it can be used not just to author printed books and articles, but also to create reusable sections or "topics" that can be combined and presented in any form. These sections can be delivered as web pages for an LMS, as a series of slides in a slide presentation for classroom or as a set of help topics.

Although DocBook has become a widely used standard, there are other systems and markup vocabularies that provide capabilities similar to what DocBook provides. But DocBook provides several features that have made it a big success:

- Open Source Technology
- A full range of freely available Authoring Tools
- Specification thoroughly documented
- It has been widely used and tested by organizations such as Sun, Microsoft, Hewlett-Packard, Novell, SCO, Caldera, and Red Hat, CERN, or by open-source groups KDE, GNOME, and the Linux documentation Project.
- Multiple output formats can be easily obtained using free tools along with the DocBook stylesheets (Stayton, 2005), which can be found at the DocBook Open Repository (<http://docbook.sourceforge.net>).

Occasionally the term DocBook is also used as a catchall term to describe the markup language and the tools used to process and convert DocBook documents into other formats. Technically, DocBook is only the DTD, but DocBook isn't quite as useful without the tools.

To summarize, DocBook can be used to realize the full potential of an XML-based publishing system:

Content Reuse: by separating form from content, the goal of any XML publishing solution is to simplify the reuse and republishing of information in any presentation format.

Sharing: the objective of XML is to enable the construction of information in a media-neutral way so that it can be easily exchanged, regardless of authoring tools or proprietary formats.

Relevance: like XML languages, DocBook, enables the creation of content in modules that can be automatically assembled based on specific needs, so that individuals get only and all the information they need.

Automation: The key to improving content reuse, sharing, and relevance in a cost-effective manner is automation. XML is a key point in enforcing the consistent structure that automated processes require.

Software can be written for XML-based systems to automate simple tasks such as validating entries, calculating values, sorting and similar operations. An example of this is "automatic page composition."

Often writers will continually adjust individual page layouts as they compose documents. Obviously this can take up a great deal of time. One solution to this problem is to create software that lets the computer make the page layouts while letting the author concentrate on creating new content.

The main selling point for DocBook is its portability. A document written in DocBook markup can be converted into HTML, PostScript, PDF, RTF, DVI, and plain ASCII text easily and quickly without the need for expensive tools.

The syntax of a DocBook document is wholly contained in the simple rules of XML markup and in the DocBook DTD inherent in every DocBook document. The semantics are slightly less distinct. For example, the DTD contains certain semantic features that determine which elements can or must occur inside other elements. The DocBook tags are applied so that they have a certain "common sense" semantic content, at least to English speakers. But other more detailed semantic issues rely on specific publication guidelines, common usage rules, and editorial judgments (for example, those governing the type of list appropriate in a certain place in the text).

3. USING DOCBOOK TO AID IN THE CREATION OF LEARNING CONTENT

As we have seen in the last section, DocBook only proposes a vocabulary and we need to associate a semantic to the DocBook tags by using the DocBook Definitive Guide (Walsh, 1999) or we need to create a style guide that could be shared among a set of authors. Now we are going to show our view of how DocBook can be applied to learning content to ease this task.

Prior to imparting a course, a teacher generally starts with a definition of what the goals of the course are going to be (*figure 2*, step a)). These goals

may be described as an informal narrative description, but he/she can also formalize these goals by defining the learning objectives that cover them.

Once these objectives are clear, the teacher's work continues with a phase of information gathering using web search engines, books or even searching a LO repository such as the Multimedia Educational Resource for Learning and Online Teaching (MERLOT) initiative (McMartin, 2004) (*figure 2*, step b)). This phase can be facilitated by DocBook because there are tools that can apply "up conversion" from a number of formats (e.g. HTML, plain text, Texinfo, and OpenOffice). The author can convert a number of documents to the DocBook XML format. These documents will form the initial repository of raw learning content.

Following the workflow depicted in *figure 2*, once a teacher has collected a critical mass of information in his/her raw content repository, the next step is to distil this raw content into initial LOs creating summaries of the content, describing the highlights and extracting the main concepts included (*figure 2*, step c)). The information generated at this step will be a part of the metadata information of the LOs.

This stage requires teachers to deal directly with DocBook and XML. This task can be facilitated using XML Authoring Tools such as Altova XML Spy, Emacs, etc. These tools will help authors to generate a valid DocBook document using a text-based approach. It is also possible to use What You See Is What You Get (WYSIWYG) editors such as Altova Authentic, or other graphical solutions such as Lyx and OpenOffice.

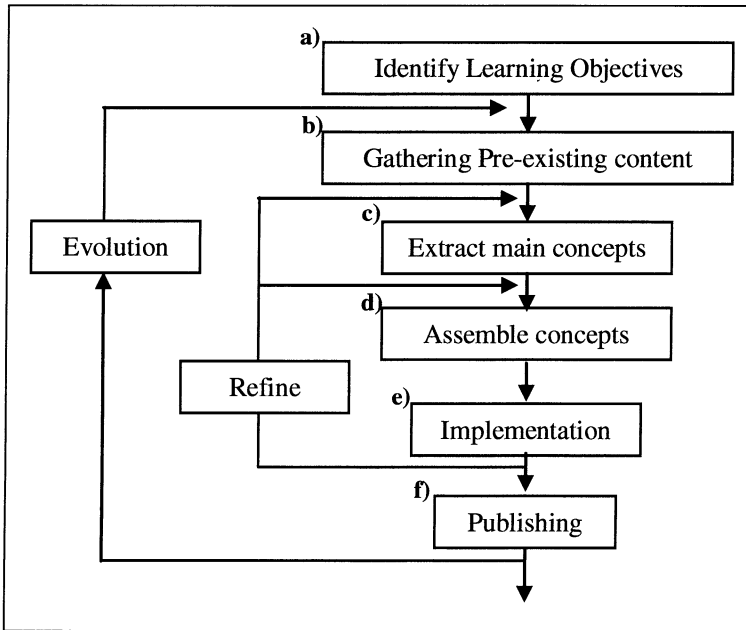


Figure 2. Learning content creation workflow

In addition, the teacher must decompose the raw content into small and manageable LOs. These small LOs can be classified according to the previous abilities necessary to understand them. Thus, when interoperating content, the possibility of finally using these LOs can be assessed according to the real level of the target learners.

Another reason for decomposing large LOs is to make them more flexible (i.e. reusable). How big a LO should be or how much of the LO can be decomposed (i.e. the granularity of the LOs) are key points for LO reusability (Polsani, 2003).

Once the teacher has distilled the LOs, the next step requires their careful assembly to meet the proposed learning objectives (*figure 2*, step d)). How these LO are aggregated depends on the instructional design that the teacher wishes to apply. Taking into account the expected educational abilities of the target learners is obviously necessary as well.

During the implementation stage (*figure 2*, step e)) using DocBook, the teacher deals with the main part of content codification, for example, the creation of a glossary for each LO, the use of admonitions for explanatory annotations of obscure parts of the content, the inclusion of media content, paragraph divisions for coherent reading, etc.

In addition, DocBook allows for particularizations of its document model (i.e. DocBook DTD) in order to adapt it to our specific needs. For instance, we can omit some tags: this would reduce complexity and still be fully DocBook compatible.

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook V4.2//EN"[
<!ENTITY subdoc1 SYSTEM "sect1.xml">
<!ENTITY subdoc2 SYSTEM "sec2.xml">
]]>
<article xml:lang="en">
<title>Content Aggregation</title>
  &subdoc1;
  &subdoc2;
<para>Additional content</para>
```

Figure 3. LO aggregation using XML entities. We can use the Entity mechanism of XML as a macro expansion facility to include external content.

At this point, the teacher has done all the work: he/she has implemented his/her instructional design with LOs. With a defined information structure, and a means to edit and maintain their objects, teachers will need to establish and develop methods to deliver the information to the learners, that is to say, to publish the content itself (*figure 2*, step f)). The power of DocBook is that any and all delivery methods a teacher may choose (PDF, HTML...) can be created from the same information-object repository without major restructuring and rewriting of the information.

In addition, DocBook is built in such a way that it can be used to author any kind of document or modular content. This means that it is possible to use it beyond authoring printed books and articles, thus creating reusable sections for a Web site or a series of slides in a slide presentation to be used as guide in the classroom, etc.

Finally, there are two stages that imply backwards cycles. A Refine stage may be necessary during the course to modify the aggregation of LOs in order to deal with the learners' real abilities (as opposed to the initially expected abilities). The Evolution stage, as its name suggests, allows the teacher to develop the course further after gathering feedback from the learners or using his/her own experience while teaching the course.

3.1 Limitations of DocBook and Related Projects

The first problem that may arise using DocBook is its XML syntax. Using XML syntax is not as comfortable as working with word processors, so some training for the content authors is needed together with the cooperation of technicians to support them using XML related tools. It is also possible to make use of graphical tools to minimize the jump between word processors and XML.

Another problem is DocBook vocabulary expressiveness. Since we have invested a great effort on XML, we want to maximize the usefulness of XML but DocBook vocabulary may not cover our needs. If this is the case, we can make use of DocBook DTD modularity and extend the vocabulary to cover our project needs. For example, if our project deals with literary and linguistic texts we should consider using the XML vocabulary proposed by the Text Encoding Initiative (Burnard, 1997).

4. USING DOCBOOK FOR THE CREATION OF IMS COMPLIANT LEARNING CONTENT

In this section, we present our contribution addressing the creation of complex XML files in order to work with the IMS Content Packaging specification and the associated metadata.

4.1 IMS Content Packaging and IEEE LOM Metadata

The IMS Content Packaging Specification (IMS CP, 2004) provides the functionality to describe and package different learning materials (a learning object alone, an individual course or a collection of courses) into an interoperable, distributable package that may be interchanged between LMS, LO repositories or teachers that use authoring tools compliant with this specification. Content Packaging addresses the description, structure, and format of a specific online learning interchange unit called Package Interchange Format (PIF).

According to this, a package is formed by a set of physical archives with the contents and a manifest. This manifest is an XML document that reflects comprehensive information about the package, the structure of the contents, their types and their possible organizations. More precisely the manifest contains:

- Metadata information about the package: This metainformation follows the Learning Object Metadata (LOM) specification defined by the IEEE LTSC (LOM, 2002). LOM is also used to convey the metadata associated with the other elements in the manifest (resources, organizations, and submanifests). The purpose of metadata annotation is to facilitate the obtainment of accurate search results over LO repositories/databases.
- The description of the package's resources: In its simplest form, a resource is associated with a physical archive with learning content. Actually, a resource can be associated with a number of files: A main file and a number of secondary files related to it (Like an HTML document and its images).
- The *organizations* of the resources: They are tree-like structures and their nodes, called *items*, contain references to resources. Therefore, an *organization* provides a tree based hierarchical view of the resources of the package (and thus, of its physical files). It should also be noted that a manifest can include several organizations, each one providing an alternative way to organize the contents, and therefore a different view of the package.
- The *submanifests*. A manifest can contain other simpler manifests that in turn exhibit the same structure outlined here.

4.2 Automation task of creation XML files

The problem with this model is that content creators are forced to focus their attention not only on the creation of the content, but also on the creation of LOs, tagging them with metadata and then, finally, the creation of the manifest file. In addition, the author must deal with the complex XML syntax of IMS Content Packaging or use an external tool like Reload Editor (<http://www.reload.ac.uk>) for this purpose. However, at this point we can benefit from our investment in XML.

We have developed a new module of DocBook XSLT stylesheets. This module not only generates output content (HTML), but also generates the manifest. The resulting manifest has only one organization and its tree-based organization is created using hierarchical content tags (see section 2) as items of the resulting manifest. This means that we map section, glossary and bibliography tags to manifest items.

This mapping satisfies our needs, but obviously may not fit the objectives of another project or platform. Again, it is not necessary to modify our DocBook documents. It is only necessary to change the matching rules defined in the XSLT transformation stylesheets to obtain the desired result.

In addition, DocBook defines tags to embed metadata inside documents. As long as the author provides metadata in DocBook syntax, we turn DocBook metadata tags into their equivalent LOM Metadata tags.

As we are conscious that making metadata is a tedious and error prone task, we should automate this task by providing the author with tools that can infer the metadata information using the internal information in the document. For example, if a DocBook document includes mathematical equations made with MathML, we can generate a <technical> metadata tag stating that the learner browser should be capable of rendering MathML. In addition, if we use a versioning system such as CVS, we can generate revision history metadata from information retrieved from these kinds of tools.

To sum up, after creating a set of LOs using DocBook, we use the technologies and tools associated with XML to facilitate the creation of a PIF archive and its special components, that is to say, a zip file, which includes all the necessary information that must be deployed in a LMS compliant with the IMS Content Packaging specification.

4.3 Profiling and Personalized Content

Profiling is the term used in DocBook to describe conditional text. Conditional text allows us to simultaneously produce more than one version of a document when the versions differ in minor ways. We can see how conditional text is defined by observing the **userlevel** attribute. This attribute belongs to the set of common attributes for all DocBook tags. DocBook DTD does not define what values are valid for this attribute, so we can define our own vocabulary.

A common use is to introduce values describing the level of experience required by a certain section here. Since this is an XML file, it would be very easy to manipulate the file to suit a specific user with a specific level of experience. We could filter out all the sections that are beyond the level of the learner's expertise.

Such modifications of XML content do not need to be restricted to content itself. It would be possible to apply a set of different style sheets to a DocBook document in order to adapt it to a variety of learning styles and environments.

5. CONCLUSIONS AND FUTURE WORK

Adopting XML as a preferred document format should drastically reduce our document processing costs and yield greater control over the creation, delivery and appearance of () documents. In addition, adopting DocBook provides a large set of tools that provide document management and processing support not available for HTML or word processor formats.

Our future research has to do with improving the automation of tasks, in particular with the generation of JavaScript needed by SCORM based content.

Recently the OASIS organization has promoted a new system for document creation and management called Darwin Information Typing Architecture (DITA) as an OASIS Standard. It embeds content reuse into the authoring process by using the concept of topic, and by introducing an authoring model focused on the reuse of information at the topic level. As future work we want to apply this topic-oriented approach to building personalized learning paths (Farrell, 2004) using user preferences and prior knowledge.

ACKNOWLEDGEMENTS

The Spanish Committee of Science and Technology (projects TIC2001-1462 and TIN2004-08367-C02-02) has partially supported this work.

REFERENCES

- Burnard, L. 1997, Guidelines for Text Encoding and Interchange (C M Sperberg-McQueen)
- Coombs, J., Renear, A., DeRose S., 1987, Markup Systems and the Future of Scholarly Text Processing (Communications of the ACM, Volume 30 , Issue 11) pp 933-947
- Dougherty, D., 1999, The Making Of DocBook DTD (O'Reilly XML.com at <http://www.xml.com/pub/a/1999/10/docbook/docbook-making.html>).
- Downes, S., 2001, Learning Objects: Resources For Distance Education Worldwide (International Review of Research in Open and Distance Learning. Retrieved April 25, 2005: <http://www.irrodl.org/content/v2.1/downes.html>)
- Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Wentland Forte, M., Ebel, N., Macowicz, M., Warkentyne, K., and Haenni, F., 2001, The ARIADNE Knowledge Pool System, Communications of the ACM, May 2001, Vol. 44, Nr. 5.

- Farrell, R., Liburd, S., Thomas, J., 2004, Dynamic Assembly of Learning Objects (Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters 2004), pp 162-169
- Goldfarb, C.F., 1981, A Generalized Approach to Document Markup (Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation) pp 68-73
- IMS CP, IMS Content Packaging Specification v1.1.4, 2004 (Retrieved April 19, 2005: <http://www.imsproject.org/content/packaging/>)
- Khare, R., Rifkin, A., 1997, XML: a door to automated Web applications (IEEE Internet Computing, IEEE, Volume 1, Issue 4, July-Aug. 1997) pp. 78-87
- Koper, E.J.R., 2003, Combining re-usable learning resources and services to pedagogical purposeful units of learning. (A. Littlejohn (Ed.), Reusing Online Resources: A Sustainable Approach to eLearning London: Kogan Page.) pp 46-59.
- McMartin, F., 2004, MERLOT: A Model for User Involvement in Digital Library Design and Implementation (Journal of Digital Information, Volume 5 Issue 3, Article No. 293, 2004-09-09)
- Polsani, P., 2003, Use and Abuse of Reusable Learning Objects (Journal of Digital Information, Volume 3 Issue 4, Article No. 164, 2003-02-19)
- Stayton, B., 2005, DocBook XSL: The Definitive Guide Third Edition (SageHill Enterprises)
- LOM, IEEE 1484.12.1-2002 Learning Object v1 Metadata Final Draft, 2002, (Retrieved April 18, 2005: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf)
- Walsh, N. and Muellner, L., 1999, Docbook The Definitive Guide (O'Reilly, Sebastopol, C.A, USA).
- W3C: World Wide Web Consortium, 1999, Web Content Accessibility Guidelines 1.0. Retrieved May 1, 2005: <http://www.w3.org/TR/WCAG10/>.
- W3C: World Wide Web Consortium, 2001. Extensible Stylesheet Language (XSL). Retrieved April 19, 2005: <http://www.w3.org/Style/TR/xsl/>
- W3C: World Wide Web Consortium, 2001b, XSL Transformations (XSLT). Retrieved April 19, 2005: <http://www.w3.org/Style/TR/xslt/>
- W3C: World Wide Web Consortium, 2004, Extensible Markup Language (XML) (Third Edition). Retrieved April 19, 2005: <http://www.w3.org/TR/REC-xml/>