Using DocBook and XML Technologies to Create Adaptive Learning Content in Technical Domains[1]

Iván Martínez-Ortiz [1], Pablo Moreno-Ger [2], José Luis Sierra-Rodríguez [2] and Baltasar Fernández-Manjón [2]

[1] Centro de Estudios Superiores Felipe II. Aranjuez, Spain
[2] Universidad Complutense de Madrid. Madrid. Spain
e-mail: imartinez@cesfelipesegundo.com, {pablom, jlsierra, balta}@fdi.ucm.es

Abstract

This work presents an XML-based authoring methodology that facilitates the different tasks associated with the development of standards-compliant e-learning content development. The idea is to support a unified authoring process of adaptive e-learning content by adopting the *manual writing metaphor*. According to this metaphor, which is specially well-suited for technical domains, instructors are compelled to view e-learning material as the *manuals* normally written and used to teach a particular subject. The methodology is supported by the DocBook document model (an XML application) which has a number of tools associated that facilitate the implementation of the methodology. Other tasks such as the packaging of contents or publishing in the Learning Management System (LMS) can be automated from these manuals.

Keywords: XML, Adaptive learning, e-Learning, interoperability, profiling, adaptation.

1. INTRODUCTION

Historically, the production and maintenance of educational contents has required great investment and the employment of multidisciplinary groups of experts. But technological evolutions (i.e. from CD-ROM to Internet technologies) or migrations to different e-learning platforms can make that content obsolete without any possibilities of reuse. In addition, these contents are usually designed as indivisible blocks and this affects reusability between different authors, projects or target audiences.
Solutions to facilitate data portability by using descriptive markup languages like the Standard Generalized Markup Language (SGML) have been proposed [6],[12]. SGML has nowadays been simplified and has evolved into the eXtensible Markup Language

[1] The Spanish Committee of Science and Technology (projects TIN2004-08367-C02-02 and TIN2005-08788-C04-01) and the regional government of Madrid (4555/2005) have partially supported this work.

(XML) [35], although the main ideas behind the original proposals still remain: (i) authors must concentrate on structuring the content, (ii) the content and processing methods must be separated, and (iii) the presentation is a kind of processing. These goals are accomplished by XML, which uses tags to structure the information (i.e actual content) and defers presentation information to a separate document: an XSLT (eXtesible Stylesheet Language Transformation) file [33], [34]. Markup technologies based on SGML and XML simplify the creation of adaptable contents. Indeed, by carefully choosing the markup for each domain and by making a smart use of stylesheet technologies, there are some new uses that are greatly simplified such as documents that can be deployed on many different platforms/devices (e.g. PDAs) and be tailored for the specific needs of multiple user profiles.

Another methodology to deal with reusability and adaptability problems is the *Learning Object Model* [10], [17]. This model is based on defining reusable pieces of content called Learning Objects (LO) that can then be assembled to form complete courses. This model permits the reassembly and adaptation of LO to suit the needs of different learner profiles. However, the success of this model depends on the proposal of standard specifications that define the process of creating, cataloguing and documenting the generated content. This standardization effort must deal with concepts such as facilitating the interchange of the LO (i.e interoperability) or the maintenance of metadata describing the possible uses of each specific unit. In other words, a content developer that follows the LO paradigm must deal with standardized content development, annotation and its packaging in specific formats. This supposes an additional workload for the author, but the employment of the aforementioned XML technologies can alleviate the overload because some of the processes can be automated or at least made more systematic.

This work presents an XML-based authoring methodology that facilitates the different tasks associated with standard compliant e-learning content development in technical domains (e.g. computer science). The idea is to support a unified authoring process of e-learning content by adopting the *manual writing metaphor*. According to this metaphor, instructors are compelled to view e-learning material as the *manuals* normally written and used to teach a particular subject. The methodology is based on DocBook (an XML *application*; i.e. a markup language defined with XML), which has a number of associated tools that facilitate the implementation of the methodology. Besides, DocBook also includes mechanisms that facilitate the adaptation of the manuals to different audiences and user profiles in a very simple and straightforward way. The approach described in the work can be very valuable for practitioners interested in facilitating the acceptance of e-learning standards by instructors, since the manual writing metaphor is less knowledge demanding for people without a prior depth knowledge on data management and/or computer science than the naked e-learning standards. Therefore, the main contribution of this paper is to propose an authoring approach for adaptable and reusable learning material more user-friendly for authors than the cited standards, including the production of standard and metadata compliant reusable learning objects.

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 automatically create content ready to be deployed in a Learning Management System (LMS) compliant with IMS Content

Packaging specifications. In section 5 we discuss some related work. Finally, some conclusions and future work are depicted in section 6.

2. DOCBOOK

DocBook is a markup language 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 can be made using that DTD. DocBook DTD defines a vocabulary particularly well suited to books and papers about computer hardware and software, but it is not limited to that use [30].
DocBook was originally created as a SGML application developed by O'Reilly and HaL Computer Systems in 1991 with the objective of facilitating the interchange of software manuals between companies [9]. It has been widely used and tested by organizations such as Sun, Microsoft, Hewllet-Packard, Novell, SCO, Caldera, Red Hart, CERN, and the Linux Documentation Project. As its popularity grew, becoming a *de facto* standard, the Organization for the Advancement of Structured Information Standards (OASIS) created a Technical Committee (TC) engaged in developing the suite of DocBook specifications[2]. Currently DocBook is a fully XML-compliant application.

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

Figure 1. A fragment of a DocBook Document

DocBook brings to content authoring all the benefits of XML-based descriptive markup [6], [12]. DocBook annotates the semantics of the documents, not their appearance. Using DocBook tags, authors describe the document structure (Figure 1). This focus on document structure contrasts with the more common approach of word processors, where authoring is focused on the aspects of presentation (e.g. font, size, etc). On the contrary, the presentation of DocBook elements is not defined, defining instead the conceptual meaning of the text. Actually, presentational decisions should be made by graphics designers, not content authors. This philosophy can exploit the full potential of an XML-based publishing system:

- *Content attribution.* DocBook's descriptive markup lets authors work with element types systematically. For example, in creating a glossary of foreign phrases in a document, an author could simply search for all occurrences of the tag `<foreignphrase>`. With a word processor, she would have to use the less

[2] http://www.oasis-open.org/committees/docbook/

effective and more error-prone method of searching for all phrases marked in italics.

- *Content reuse and adaptation*. The separation of presentation and content enables the reuse of the information and its repurposing in multiple presentation formats, leading to a single sourcing model of content management as depicted in Figure 2. Besides, DocBook markup provides means to include information about the intended readers in a very simple way, which can be subsequently used to adapt the DocBook marked contents to such audiences.
- *Content sharing*. DocBook documents can be constructed in a media-neutral way so that they can be easily exchanged, regardless of authoring tools or proprietary formats.
- *Content modularization*. DocBook enables the creation of content in units that can be automatically assembled based on specific needs, so individuals get exactly all the information they need.
- *Support for automation*. Being DocBook and open source, widely documented, technology, a full range of freely available authoring tools and other valuable resources have been developed, which facilitates the automation of many processes involving DocBook documentation. For instance, using free tools along with the DocBook stylesheets [27] from the DocBook Open Repository[3] and related resources, DocBook content can be transformed, formatted, and published in a number of formats (e.g. HTML, PostScript, PDF, RTF, etc). These transformations can be done easily and quickly without requiring any expensive tools. In addition, DocBook being a XML application, all the existing XML processing frameworks and technologies [3] can also be applied to automate the processing of the DocBook compliant documents.

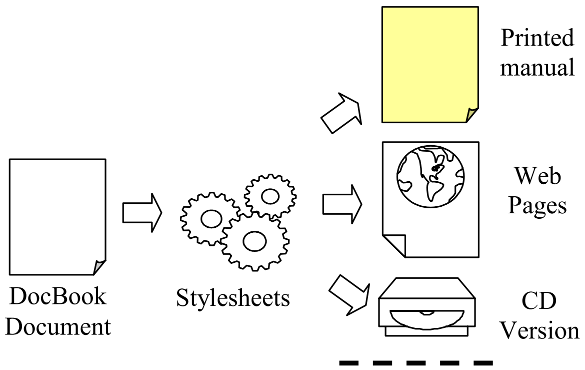

Figure 2. DocBook promotes a single sourcing model of content management.

In addition to these advantages, DocBook is highly modular and extensible. Although the DocBook DTD is very broad and complex because it considers the many options and characteristics appearing in the domain of the technical documentation, this DTD can be adapted (i.e reduced) to the specific characteristics of each particular project. The particularization of DocBook presented by Sun Microsystems called SolBook [31] is an example of reducing the syntax to fit specific projects. On the other hand, it is also

[3] http://docbook.sourceforge.net

possible to extend the language to add new tags/attributes if they are required. For instance, there are extension modules that extend the DocBook specification to support syntactic diagrams for Extended Backus Naur-Form (EBNF) notation, Scalable Vector Graphics (SVG) or the MathML equation markup language. Therefore, the specification can be tailored to accommodate specific markup needs in each application domain not contemplated by the initial specification.

3. CREATION AND USE OF LEARNING CONTENT USING DOCBOOK

Regardless of its initial orientation to the technical publishing domain, we have realized that DocBook can also be used to support an authoring process in the production of learning content in technical domains. In this process, we adopt a *manual writing metaphor*. According to this metaphor, instructors are compelled to view learning material as the *manuals* used to teach a particular subject. These manuals are then structured in accordance with DocBook. By following some additional writing rules, this process enables the automation of many different tasks (e.g. production of printouts and presentation slides, and even the extraction of potential reusable learning objects and their automatic packaging according to well-known e-learning standards). In this section we detail this use of DocBook. In subsection 3.1 we describe the authoring process. In subsection 3.2 we outline the subsequent automation during publishing enabled by this process.

3.1. The Authoring Process

In Figure 3 we reinterpret the typical workflow in the production of the learning material for a course. This workflow introduces the following stages:

- *Identification of learning objectives* (Figure 3, step a). In this stage the instructor defines the goals of the course. These goals may be described as an informal narrative description, but she can also formalize them by defining their learning objectives.
- *Gathering of pre-existing content* (Figure 3, step b). In this phase the instructor looks for the contents required to prepare the material. For this purpose, the instructor can use web search engines, books, peer support or even searching an LO repository such as the Multimedia Educational Resource for Learning and Online Teaching (MERLOT) initiative [21].
- *Extraction of the main concepts* (Figure 3, step c). Once the teacher has collected a critical mass of information from his/her raw content sources, the next step is to distil this raw content into structures creating summaries of the content, describing the highlights and extracting the main concepts included. As we will see later, part of this information can be employed as metadata when the content is structured as learning objects.
- *Assembling of the concepts* (Figure 3, step d). In this step the instructor assembles the different *chunks* of content, discarding some content and modifying some parts in order to meet the learning objectives proposed. The organization of the content depends on the instructional design that the teacher wishes to apply. Taking into

account the educational abilities expected of the target learners is obviously necessary as well.

- *Implementing the course* (Figure 3, step e). At this stage the instructor deals with the codification of the course in a suitable and homogeneous supporting format.
- *Publishing the course* (Figure 3, step f). Once the author has a well-defined information structure, and a means to edit and maintain its objects, the next step is to deliver that information to the learners, that is to say, to publish the content.

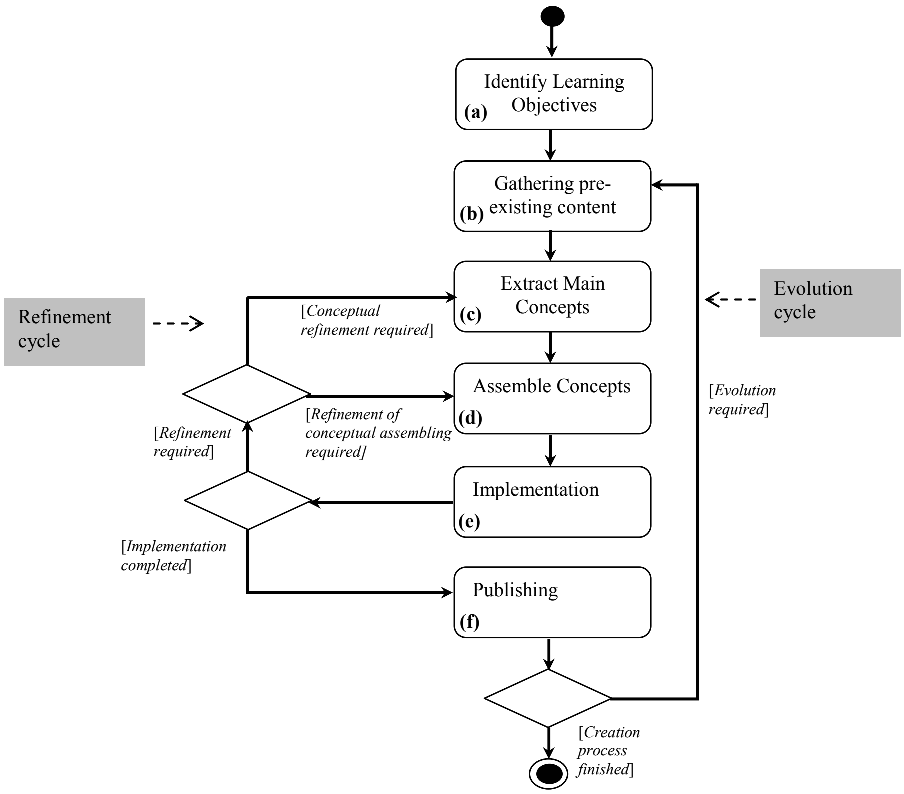

Figure 3. Learning content creation workflow

In addition, notice that there are two stages that imply backwards cycles:

- A *Refinement* cycle may be necessary during the course in order to modify the content to deal with the learners' actual abilities and previous knowledge (as opposed to initially expected levels).
- The *Evolution* cycle, as its name suggests, allows the author to develop the course further after gathering feedback from learners.

This workflow can be concreted in technical domains in terms of the *manual writing* metaphor and DocBook:

- During the *identification of the learning objectives* the instructor can also figure out the overall organization of the manual that will be produced to achieve such objectives.

- During the *gathering of pre-existing content,* the instructor can use tools for applying *up conversion* to DocBook from a number of formats (e.g. HTML, plain text, Texinfo, and OpenOffice). Therefore, the instructor can convert a number of the documents gathered to the DocBook XML format. These documents will form an initial repository of raw learning content.
- During the *extraction of the main concepts,* the instructor prepares the main structure of the manual. This stage requires instructors 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 graphics solutions such as Lyx and OpenOffice.
- During the *assembly of the concepts,* the instructor will fill the DocBook structure with the different contents that will integrate the manual. This activity can take advantage of standard modularization mechanisms for structured documents (e.g. XML entities).
- During *implementation,* the instructor will tune the fine-grain details of the DocBook codification, such as for example, the creation of a glossary for the course, the use of admonitions for explanatory annotations of obscure parts of the content, the inclusion of media content, paragraph divisions for coherent reading, etc.
- Finally, the DocBook markup added to the manual produced will permit the automation of the different tasks contemplated during publishing, including adaptation of the material to different audiences, as depicted in the next subsection.

3.2. The Publishing Stage

The use of DocBook during the authoring process described in the previous section pays off during publishing. Once the first stages of this process are completed, the instructor has a complete course in the form of XML files and associated resources such as images, diagrams or multimedia content. As mentioned before, the set of available XML tools designed for DocBook is a powerful resource that allows the transformation of this material into a variety of formats (Figure 4):

- The most straightforward publishing method is to transform all the notes into a book-like document containing the contents of the course (Figure 4a). This can be used as class notes by the teacher and, eventually, put at the disposal of learners. This transformation can be performed with available open source tools for DocBook that turn a DocBook document into a specific format. These tools (usually XSL Transformations) are heterogeneous and rich, and it is possible to generate a variety of formats in order to fit different situations. Some typical and recommended formats would be PDF, HTML (in order to post them on a website) and even RTF (to allow further modification).
- The very same DocBook document can contain additional markup that identifies the highlights of each section. These highlights can be automatically extracted to a set of slides that can be used by the teacher as a teaching aid during live lectures (Figure 4b). Although this process is not directly supported by pre-existing DocBook tools, we have implemented it using a customized XSL Transformation, turning this transformation into a process very similar to the production of the class notes.

- Finally, by following appropriate patterns in using DocBook, it is also possible to automatically extract reusable chunks of learning material (i.e. reusable learning objects, as suggested in Figure 4c) that will be subsequently reused during content gathering by the author or by a different instructor. In addition, these objects can be annotated using standard e-learning notations and packaged in standard formats (Figure 4d), which can be integrated in standards-compliant Learning Management Systems (LMS).

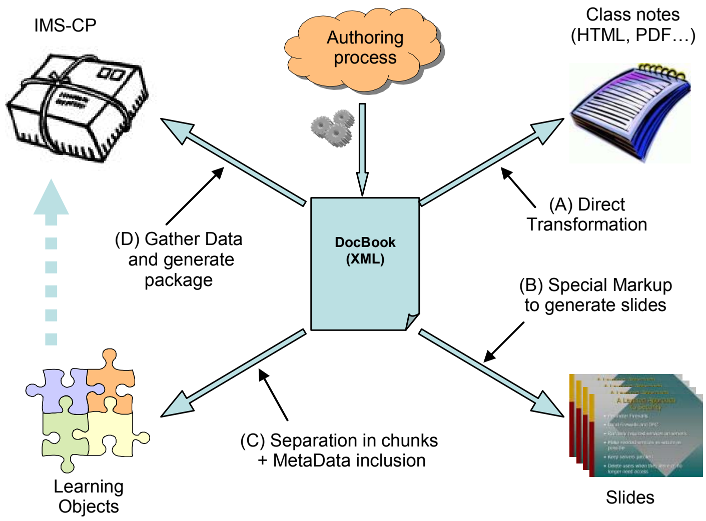

Figure 4. Publishing alternatives for learning material organized according to the *manual* metaphor and DocBook.

Among these uses, we consider that those regarding e-learning technologies and the learning object paradigm are the most interesting ones. These aspects of the methodology are further discussed in the following sections.

4. USING DOCBOOK IN THE LEARNING OBJECTS PARADIGM

The methodology described in the previous section can easily accommodate the LO model. Although the concept of LO is flexible and subject to interpretation, there are a few common concepts that are widely accepted [23]: an LO must be self-contained, interoperable and must include metadata that defines it and facilitates its assembly and interoperability. In addition, LO must be easy to adapt to each particular personalized learning experience. In the next subsections we show how our methodology addresses each one of these requirements. In subsection 4.1 we examine self- containment. The inclusion of metadata is discussed in subsection 4.2. Subsection 4.3 deals with the interoperability aspects. Content adaptation and personalization is treated in subsection 4.4. Finally, subsection 4.5 summarizes how all these aspects are integrated into a coherent framework.

4.1. Addressing Self-Containment

In our methodology, LOs are naturally mapped onto DocBook sections. This rhetorical constraint must be tackled by instructors during the definition of the objectives of the course and the formalization of the instructional design in accordance with the *manual writing* metaphor. Indeed, in order to represent a meaningful LO, each section must have a well-defined pedagogical goal. This constraint fits well in the metaphor, leading to well-structured manuals where each topic is located in a specific section. In addition, composite LO can be represented as sections that enclose subsections. Therefore, when the instructor decomposes the pool of raw content obtained after the initial stages into small and manageable LO, he/she is implicitly deciding the main structure of the manual in terms of sections and subsections. Therefore, the granularity of the LO (i.e. the exact size of a LO or how much of the LO can be decomposed), which is a key point for LO reusability and a subject of continuous discussion [23], is translated in the nesting levels of the resulting DocBook document.

Notice that the map established between LO and DocBook sections is a satisfactory solution for a technical viewpoint, since:

- XML, and therefore DocBook, documents can be physically organized in terms of *entities* [29]. This sort of referential markup allows the introduction of a large piece of XML content into another document. The parent document thus becomes an aggregation of the documents embedded. Hence in our approach aggregation of self-contained learning objects is carried out by using the entity mechanism in a straightforward way (Figure 5).
- A document can be automatically decomposed in reusable document fragments by using a suitable transformation, which can be specified in a declarative and standard way using the XSLT language. We have defined an XSLT transformation that splits any DocBook document in terms of its constituent sections.

```
<!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>Aditional content</para>
```

Figure 5. LO aggregation using XML entities. We can use the entity mechanism of XML as a macro expansion facility to include external content.

4.2. Inclusion of Metadata

The metadata for the LO would be useless unless it could be easily read and interpreted by anyone. The final purpose of LO annotation with metadata is to facilitate authoring and assembly tasks by providing a mechanism to obtain accurate search results over LO repositories/databases. This is the reason why there are different standardization proposals available. One of them is the IEEE Learning Object Metadata (LOM) standard [20], which provides an information model that defines the structure of a metadata instance for a LO. A metadata instance describes relevant characteristics of the

LO grouped into general, life cycle, meta-metadata, educational, technical, rights, relation, annotation and classification categories.

The jargon (and the sheer size) of the IEEE LOM specification can easily overwhelm an author unless he/she is a specialist in the field of interoperable content. Instead of forcing the author to learn such complex specifications (which may even be superseded by other newer specifications), the DocBook methodology allows a simple and specification-independent mechanism to add metadata to the content.

To begin with, the DocBook format includes syntax for basic metadata. In addition, this syntax can be enhanced using the DocBook DTD extensibility mechanism in order to fit specific needs. The result is a set of metadata embedded in the XML document. These metadata can be automatically interpreted and transformed to the specification desired without bothering the author. Again, this has been implemented using a XSL Transformation specifically designed to produce IEEE LOM compliant metadata, but it would be possible to code a new XSL Transformation to support a different specification without changing the DocBook documents.

4.3. Interoperability

Content interoperability requires the content to be packaged in standardized manners in order to be deployed in a system that complies with those standards. The IMS Content Packaging Specification [16] provides the functionality to describe and package learning materials (such as an individual course or a collection of courses) into an interoperable, distributable package. The Content Packaging Specification addresses the description, structure, and location of online learning materials.

In order to create a standardized package of interoperable LO that can be displayed in a web based LMS, we need the LO themselves (ideally in XML format), their associated media files, the descriptive metadata and the manifest put together in a single entity called PIF (Package Information Format) which is basically a zipped file containing all the necessary files (Figure 6a).

The structure of IMS packages is depicted in Figure 6b. According to this, a package is formed by a set of physical archives with the actual contents and a manifest. This manifest is a XML document that reflects global information about the package, the structure of the contents, their types and their possible organizations. More precisely, the manifest contains:

- A section of *metadata* summarizing global (meta)information about the package.
- The description of the *resources* of the package. In its simplest form, a resource is associated with a physical archive with learning content and some metadata. It is also possible to describe resources associated with a main file and a set of secondary files. This makes it possible to bundle content sets like a main HTML file and the images related to it.
- The *organizations* of the resources. Each organization represents a tree structure whose nodes can refer to resources. The nodes in this tree are called items and they contain a reference to their corresponding resources using the unique identifiers of the resources. Therefore, an organization provides a tree based hierarchical view of the resources of the package (and thus, of its physical files).
- The *submanifests*. A manifest can contain other, simpler manifests that in turn exhibit the same structure outlined here.

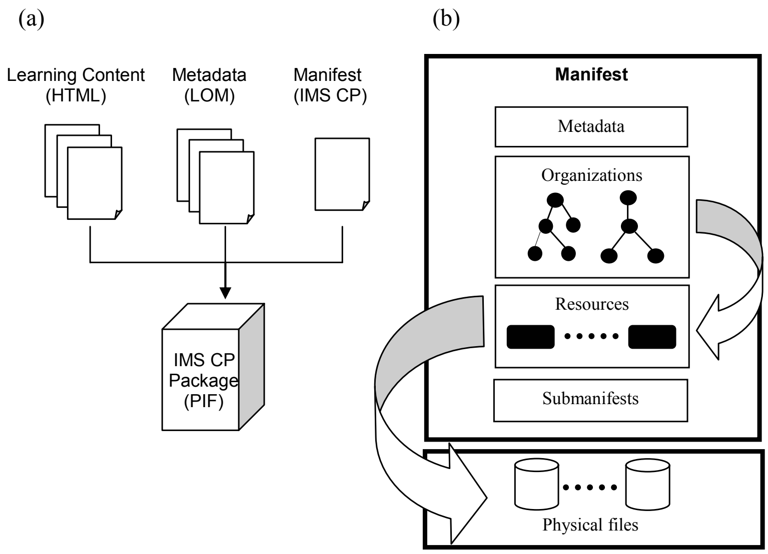

Figure 6. (a) Building a distributable package; (b) Structure of an IMS package.

It is interesting to note the parallelisms between these notions and the concepts explained above. If the DocBook files follow the aforementioned guidelines, it is possible to automatically create the corresponding IMS manifest. The submanifests provide support for LO aggregation, the instructional design becomes an organization and the embedded metadata is adapted to the desired specification and included in the manifest.

4.4. Profiling, Adaptation and Personalized Content

Profiling is the term used in DocBook to describe conditional text. Conditional text allows us to produce more than one version of a document, and the versions differ in minor ways. Conditional text can be implemented in terms of the *userlevel* attribute. This attribute belongs to the set of common attributes for all tags of DocBook. DocBook DTD does not define what values are valid for this attribute, so each user or community can define its own vocabulary.

```
<section userlevel="beginner">
  <para userlevel="intermediate">
    … Paragraph content goes here …
  </para>
  <para>
    … Paragraph content goes here …
  </para>
</section>
```

Figure 7. DocBook document with user detail levels.

We have used conditional text mechanisms to cope with content personalization, as exemplified in Figure 7. The cited *userlevel* attribute is the hook where adaptation and personalization mechanisms can be anchored. This enables a great flexibility:

- Adaptation and personalization can be carried out on the basis of the learner's expertise, as in Figure 7. Sections beyond the level of the learner can be filtered out.
- On the other hand, adaptation and personalization could be carried out according to a pedagogical goal. For instance, the instructor may want to provide two views of the learning content, one view with only the exercises and another with only theoretical components. Provided a document is appropriately described, any view of the document content can be published independently.

From an implementation point of view, personalization is carried out by an appropriate XSLT transformation that filters out the unnecessary elements.

4.5. Putting It All Together

In Figure 8 the process for creating standardized packages of interoperable and personalized LO that can be displayed in a web based LMS according our approach is depicted. According to this process:

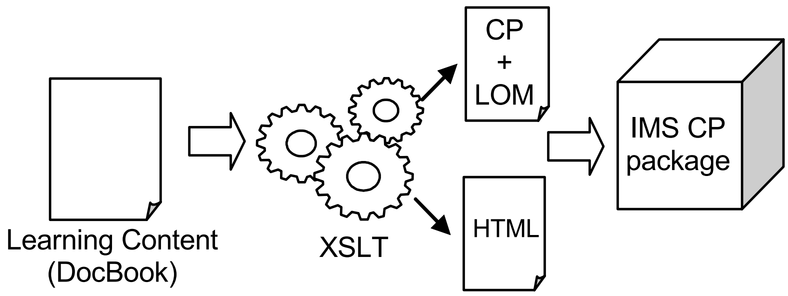

Figure 8. Building personalized learning content using DocBook

- The instructor provides the DocBook manual that aggregates the different LO, together with the associated media files.
- At this point all the benefits from the investment in XML technologies emerge (and their associated work overload both from the instructors' and the XSLT developers' perspectives). If the manual has been authored correctly, the available XSLT transformations can personalize LO according to appropriate adaptation and personalization criteria, present the personalized LO in HTML, extract their metadata to a standardized format and organize the LO in a manifest (Figure 8).

We have used our experimental IMS-based LMS <e-Aula> [26] as a testing workbench for our approach. In Figure 9 we sketch the deployment in <e-Aula> of an IMS CP package automatically generated from a DocBook document with a course about the Unified Modeling Language (UML).

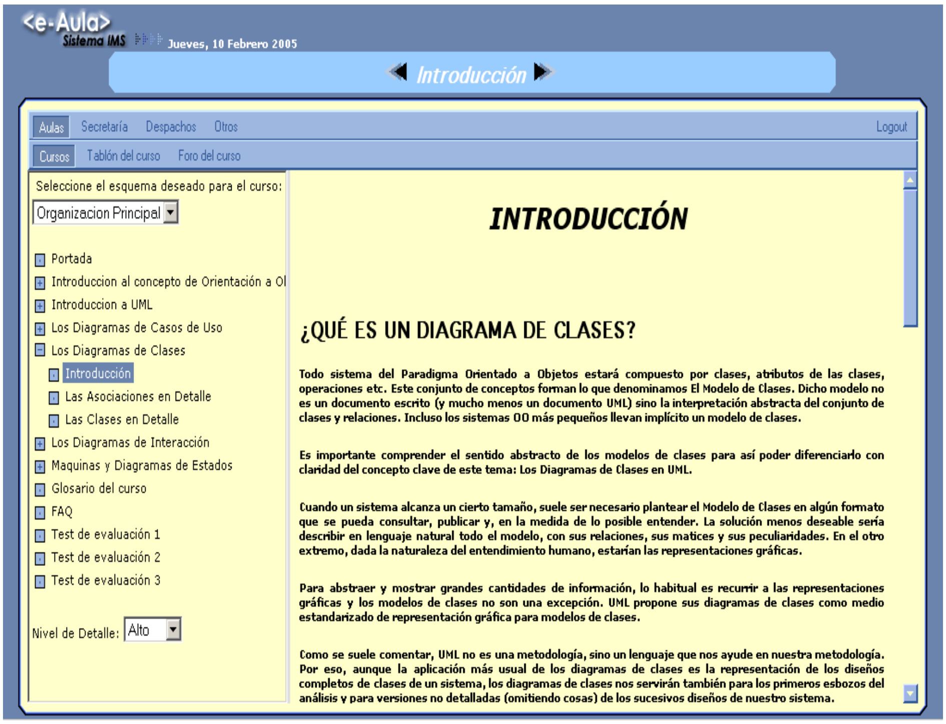

Figure 9. DocBook document transformed, packaged and deployed in <e-Aula>

5. RELATED WORK

The single-source model in the production of learning content is analyzed in several works. In [22] a DocBook-based system developed for presenting a post-graduate module in object-oriented programming at Dublin City University is presented. This work highlights the advantages of the approach regarding user profiling, accessibility, extensibility, content protection, and support for the generation of interactive content. An extensive case-study experience is reported in [28], where a XML-based markup language inspired in DocBook is used in the development of the International Baccalaurate's online Economics Subject Guide. Nevertheless, these works do not report of any special attempt to produce standard-compliant reusable learning objects.

DocBook was proposed as the content markup language in EML [18] (Educational Modelling Language), the main precursor of IMS Learning Design (IMS LD) [19] (specification that leaves open the content structuring format). DocBook has been also used in the production of learning contents in several projects related with the implementation of advanced e-learning technologies based on open-source solutions. One of these projects is Edukalibre [14], [15] which promotes a methodology for the production of learning materials similar to the adopted in communities devoted to the development of free/open software. Being widely used in those communities for documentation purposes, DocBook is the Edukalibre base format. All the other document formats are therefore transformed into DocBook documents. E-lane (Europe and Latin America New Education) [8] is a project oriented to the development of an open-source e-learning platform and low cost high quality educational material for the Latin America society. In E-lane, DocBook is also adopted as a basic tool for the edition of contents. In this case, the use of DocBook is mainly oriented to the production of educational contents. This use is also highlighted in ALF (Active Learning Framework) [5], an educational framework providing several Computer Supported Collaborative Learning facilities and scheduled scenarios of learning activities. In this work we go a step further by using the language also as a tool for producing LO adapted to the specific needs of particular learners and for packaging them according with well-established e-learning *de facto* standards.

In [13] a DocBook-based implementation of SeLeNe (Self eLearning Network) is proposed. SeLeNe is an architecture for metadata management and composition of learning objects. In this implementation, a similar approach to the described in the present paper is used to map DocBook documents onto LO. Indeed basic LO are identified with `section` elements and composite ones with `chapter` and `book` elements. Nevertheless, this implementation is oriented to the indexing and categorization of learning objects in order to support their subsequent retrieval and composition. The descriptive nature of DocBook is also exploited in [1] with the definition of an educational ontology that can be used for organizing the learning information and for satisfying the demands of Intelligent Learning Management Systems.

Decomposition of a document in smaller reusable units in terms of its structure resembles the work carried out in the context of *slicing book technology* [7]. The processing of DocBook documents proposed in this paper can be considered as simplified particular case of such a technology. Similar ideas are also used in [2], where the structure of the documents is exploited in their reuse for the production of modular courses based on the LO paradigm. In this proposal, which is inspired in the construction kit metaphor described in [4], documents are mapped onto a common document abstract model, and this representation is subsequently processed for producing basic and composite learning objects. Contents in the basic objects are further structured in terms of a XML-based domain-specific markup language called BrickML. The resulting objects can be used in the Lyssa authoring system [2], and they can be also translated into a SCORM-compliant representation using packaged transformation specifications. While this approach can deal with arbitrary document formats by providing suitable loaders, it is substantially more complex than the methodology described here.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described the *manual writing* metaphor, an authoring metaphor that has proved useful as an authoring methodology of learning materials in technical domains. This metaphor has been implemented using DocBook, allowing us to take advantage of a huge amount of pre-existing supporting tools, as well as of other general-purpose XML processing frameworks and technologies. Since it is a XML-based descriptive markup language, DocBook facilitates the retargeting of the content by producing book-like printouts in many different formats. Also, the adoption of additional writing guidelines enables the automatic production of classroom slides, and promotes content adaptability.

Our solution can easily accommodate the ongoing standardization efforts in the e-learning arena. We have proposed a set of writing guidelines, according to which each DocBook section is identified with a LO. By doing so, the standard XML entity mechanism can be used to assemble LO, while suitable XSL Transformations can be used to extract the LO embodied in a manual. In addition, standard XML technologies can be used to pack all the LO according to the IMS specifications, and the resulting packages can be incorporated into IMS-compliant learning management systems.

Regardless of the advantages cited, the methodology still relies intensively on the use of XML syntax. Writing XML is far from being as comfortable as working with word processors. This means that the author requires additional training and the cooperation of technicians to learn and use XML related tools. While in technical domains this drawback is often irrelevant (due to the author's skills), the generalization of the approach to other domains can require further support. For this purpose, the use of graphical tools may reduce the gap between using word processors and the XML authoring methodology.

Although the DocBook vocabulary was initially designed for technical documents, it is possible to make use of the DocBook DTD modularity mechanisms to extend the vocabulary to cover the needs of projects in different domains (e.g. if the project deals with literary and linguistic texts the XML vocabulary proposed by the Text Encoding Initiative [24] could be considered as a feasible extension). However, most of the benefits of this methodology are a consequence of XML in general, not DocBook in particular. It would be possible to choose alternative formats more suited for other domains and still retain all the benefits of this methodology. We also contemplate the possibility of exploring alternative XML technologies. For instance, the OASIS organization has promoted as an OASIS standard a new document creation and management system called Darwin Information Typing Architecture (DITA). DITA builds content reuse into the authoring process using the concept of topic and, as future work, we want to apply this topic-oriented approach to the definition of personalized learning paths [11] using user preferences and prior knowledge. Alternatively, we are also considering applying our document-oriented development approach [25] in order to yield a more generic domain-independent methodology.

Regardless of the aforementioned efforts, currently we are exploring another interesting feature enabled by DocBook markup: making the content accessible for people with disabilities. W3C proposed a set of guidelines [32] that were intended for developers of web and authoring tools, but these guidelines can also apply to DocBook documents. As a simple example, it is possible to check if the content contains alternative text descriptions for every piece of multimedia content and to generate reports informing authors of potential violations of the accessibility rules. Finally, we are also addressing the automatic generation of Sharable Content Object Reference Model (SCORM) compliant courses from DocBook manuals.

7. REFERENCES

[1] Amorim, R., Lama, M., Sánchez, E., Vila, X.A., (2003), An Educational Ontology based on Metadata Standards (European Conference on eLearning, Glasgow, UK), pp. 29-36

[2] Baudry, A., Bungenstock, M., Mertsching, B., 2004, Reusing Document Formats for Modular Course Development (Word Conference on Educational Multimedia,

Hypermedia & Telecommunications, ED-MEDIA), pp. 535-537 1. Birbeck, M. et al., 2001, Professional

[3] Birbeck, M. et al., 2001, XML 2nd Edition (WROX Press, Birmingham, UK)

[4] Bungenstock, M., Baudry, A., Mertsching, B., 2002, The Construction Kit Metaphor for a Software Engineering Design of an E-Learning System (*World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Denver, Colorado, USA), pp. 216-217

[5] Boticario J. G., Raffenne, E., Aguado, M., Arroyo, D., Cordova, M. A., Guzmán, J.L., Garcia, T., Hermira, S. Ortíz, J., Pesquera, A., Romojaro, H., Valiente., S., 2003, The INNOVA Approach (Eadtu Anual Conference: E-Bologna, Madrid, Spain)

[6] 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

[7] Dahn, I, 2002, Slicing Books - The Author's Perspective (In Bromme, R., Stahl, E (eds.), Writing Hypertext and Learning, Pergamon), pp. 125-151

[8] Dolan, D., Delgado-Kloos, C., Muñoz-Organero, M., Pardo, A., 2005, E-LANE: Open Source eLearning in Latin America (Open Source for Education in Europe, Research & Practice Conference, Heerlen, The Netherlands), pp 117-126

[9] 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)

[10] 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)

[11] 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

[12] Goldfarb, C.F., 1981, A Generalized Approach to Document Markup (Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation), pp 68-73

[13] Gueye, B., Rigaux, P., Spyratos, N., 2004, Taxonomy-Based Annotation of XML Documents: Application to eLearning Resources (Methods and Applications of Artificial Intelligence: Third Hellenic Conference on AI, SETN 2004, Samos, Greece, Lecture Notes in Computer Science 2035), pp. 33-42

[14] Gonzalez-Barahona, J.M., Dimitrova, V., Chaparro, D., Tebb, C., Mazza, R., 2005, Producing educational resources in the libre way: The Edukalibre project (IADIS International Conference WWW/Internet, Lisbon, Portugal)

[15] Gonzalez-Barahona, J.M., Tebb, C., Dimitrova, V., Chaparro, D., Romera, T., 2005, Transfering Libre Software Development Practices to the Production of Educational Resources: the Edukalibre Project (The First International Conference on Open Source Systems, Genova, Italy)

[16] IMS CP, IMS Content Packaging Specification v1.1.4, 2004 (Retrieved April 19, 2005: http://www.imsproject.org/content/packaging/)

[17] 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

[18] Koper, R., Manderveld, J., (2004), Educational Modelling Language: Modelling Reusable, Rich and Personalised Units of Learning (British Journal of Educational Technology, Volume 35), pp. 537-576

[19] Koper, R., Olivier, B., 2004, Representing the Learning Design of Units of Learning (*Educational Technology & Society*, Volume 7, Issue 3), pp. 97-111

[20] 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)

[21] 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)

[22] Molloy, D., 2003, Single Source Interactive and Printed Content Publishing Using the DocBook XML Standard (Second International Conference on Multimedia and ICTs in Education m-ICTE. Badajoz, Spain)

[23] Polsani, P., 2003, Use and Abuse of Reusable Learning Objects (Journal of Digital Information, Volume 3 Issue 4, Article No. 164, 2003-02-19)

[24] Rathz, S., Walsh, N. and Burnard, L., 2004, A unified model for text markup: TEI, Docbook, and beyond (*Proceedings of XML Europe 2004*)

[25] Sierra, J.L., Fernández-Valmayor, A., Fernández-Manjón, B., Navarro, A., 2004, ADDS: A Document Oriented Approach for Application Development (Journal of Universal Computer Science, issue 10(9)), pp 1302-1324

[26] Sierra, J.L., Martinez, I., Moreno, Lopez, J., Fernandez-Manjón, B., 2005, Building Learning Management Systems Using IMS Standards: Architecture of a Manifest Driven Approach (4th International Conference on Web-Based Learning. Hong Kong), pp. 144-156

[27] Stayton, B., 2005, DocBook XSL: The Definitive Guide Third Edition (SageHill Enterprises)

[28] Walsh, L., 2004, Using Extensible Markup Languages for the Single Source Delivery of Teaching Resources via Print and the Web: A Practical Example (Australian Society for Computers in Learning in Tertiary Education ASCILITE. Perth), pp. 913-923

[29] Walsh, N., 1998, Entities: What are They Good For ? (O'Reilly XML.com Retrieved April 25, 2005: http://www.xml.com/pub/a/98/08/xmlqna0.html)

[30] Walsh, N. and Muellner, L, 1999, Docbook The Definitive Guide (O'Reilly, Sebastopol, C.A, USA).

[31] Walsh, N., 2001, Introduction to Docbook (Winwriters Online Help Conference)

[32] W3C: World Wide Web Consortium, 1999, Web Content Accessibility Guidelines 1.0. Retrieved May 1, 2005: http://www.w3.org/TR/WCAG10/

[33] W3C: World Wide Web Consortium, 2001. Extensible Stylesheet Language (XSL). Retrieved April 19, 2005: http://www.w3.org/Style/TR/xsl/

[34] W3C: World Wide Web Consortium, 2001b, XSL Transformations (XSLT). Retrieved April 19, 2005: http://www.w3.org/Style/TR/xslt/

[35] W3C: World Wide Web Consortium, 2004, Extensible Markup Language (XML) (Third Edition). Retrieved April 19, 2005: http://www.w3.org/TR/REC-xml/