An Approach to the Use and Automatic Generation of Web-Based Learning Materials*

V. Fresno-Fernández, S. Montalvo-Herranz, J. Pérez-Iglesias,
J. Urquiza-Fuentes, and J.A. Velázquez-Iturbide

ViDo Group, ESCET - Rey Juan Carlos University, Mostoles, Spain
{v.fresno, smontalvo, joperez, j.urquiza, a.velazquez}@escet.urjc.es

Abstract. This paper presents an approach to the automatic generation and use of web-based learning materials (WLMs). It consists in three phases: contents creation, division and browsing. The use of XML in the creation phase allows achieving independence between contents and their presentation, in addition to contents reuse. The splitting phase is an automatic process whose output can be used as WLMs for different user interfaces. In the browsing phase, the interfaces are generated; we have implemented two: a user-adaptative interface and a non-adaptative one. The main features of our approach enables users to create learning contents without XML technical knowledge and also enables to effectively navigate the material to their needs.

Keywords: e-learning tools development, web-based education, teaching materials, electronic documentation.

1 Introduction

Nowadays, Internet has become the main source of information in the world and its growth results unstoppable. A huge amount of data is accessible at the Net and several technologies for extraction and sharing of information are being developed. This is not different in educational environments. In this context, the development of web-based courses and their materials is growing and it is more and more important for distance education. This new educational context exhibits important differences with respect to traditional education [1]. The way of presenting and creating learning contents must be adapted to several features [13]: *asyncronicity of teaching and learning* (students study at their most suitable time and place, and the presence of a teacher is not always needed or possible), *maintenance* (web courses can be updated at any moment, whereafter the update is directly available to all students), *alternative structures* (hyperlinked, non-linear structures as tree-like hierarchies or graphs), *external links* (tapping unlimited resources on the Internet using direct links), *integration of applications* (discussion groups, group learning, peer assessment, etc).

* This work is supported by the Spanish Research Agency CICYT under contract TIC2000-1413.

WLMs are mainly based on hyperlinked contents, functional use of interactivity and multimedia components [15]; and they must be tailored to:

1. *Contents exposition.* The contents can be divided according to either *structural* or *semantic* criteria. The structural approach divides contents based on its hierarchical structure and the semantic one divides by concepts. Examples of the structural approach can be found in [5, 11], that generates a division in HTML pages by chapters, (sub)sections or (sub)paragraphs. Examples of the semantic approach can be found in [2, 9, 14].
2. *Learning is not only reading.* Contents that require user interaction (animations, videos, audios, quizzes, etc.) can be made available [14].
3. *Interface and contents customization*, adapting tools to the users [16, 19].

WLMs are made up of static and interactive contents, but they can also be divided into those that do not allow users to modify their contents and interface [8] and those that do it [14, 13, 11]. In the first case, any web browser can be used as the user interface and in the second case, more complex interfaces are needed.

There is a second open question: are students ready to use these new environments effectively? In [1] it is stated that using an electronic environment requires skills and learning processes that are different from those used in a paper-based or face-to-face environments. Other authors claim that difficulties can be found by students adapting themselves to electronic learning methods [10, 18, 4, 12]. Consequently, the learning task can be facilitated by using tools that imitate others that are well known. The "book" metaphor and its implementation as e-books is a good approximation [17] because it allows using, in an electronic environment, a well known and common tool in classical learning. As a result, the potential confussion of hypertext [15] is overcome. One of the main features of the "book" metaphor is that the contents are structured in pages, in addition to chapters, sections, subsections, etc. To simulate a printed book, all the contents in a page will simultaneously be visible without scroll.

Who must divide the contents into pages? Some works [14, 5] consider the author in charge of creating contents that fit the page size of a specific tool. However, we think that the author must center his/her efforts in the production of learning contents and not in splitting pages. In this respect, we will make the production of contents and the division process independent and will separate the contents and its presentation. In this paper, an approach to the use and automatic creation of WLMs is presented. It is made up of three phases:

1. Contents are generated as a semi-structured document.
2. These contents are transformed into a set of pages.
3. The pages are used as WLMs, showing them as an interactive material.

This paper is organized as follows. In section 2, we briefly describe some related works. Our approach will be presented in section 3. The main technical details necessary to implement our approach are described in section 4. Finally, some concluding remarks are made in section 5.

2 Related Work

In this section, some WLMs related tools are described, focusing on the process of contents generation.

CourseGenie [3] is a commercial tool that converts a MsWord document into a set of web pages including navigation and interactive features. First, it uses its own engine to convert a MsWord document into a XML document. Then, it uses an own XSLT engine to transform the XML document into a set of interactive HTML pages. It splits without paying particular attention to the sizes of the resulting pages. EasyGenerator [6] is another commercial application. The author creates contents in separated pages. The application is made up of several tools, helping the author generating learning contents in pages, designing the course final appearance and finally, allowing students to use the course. EasyProf [7] is another commercial tool. It provides an interface for the author to edit, create and format all type of contents and it guides him/her stepwise. In [14] an application to produce electronic course books (ECBs) is presented. Such ECBs consist in a set of HTML pages of two main categories: slide pages and resource pages. The tool has an interface to insert HTML code through different menus, buttons, areas, etc. and does not limit the size of the page. In [11] a tool to manage a course book is presented. The implementation of the course includes several steps: converting the book into HTML through Adobe Frame-Maker; linking the HTML pages with the desired structure; indexing the pages by a search engine; developing interactive components and implementing user authentication and tracking. The division is acomplished by sections of a printed book. All of these tools force the user to pay attention to the appearance of the contents while they are being created. However, we think that the process of creating contents should be independent from their presentation as far as possible. Therefore, the transformation of the contents into the final WLM must be automated. Finally, all the processes must be platform independent.

3 Our Approach

In this section, we describe the different phases to achieve the independent creation and automatic generation of WLMs and their later use.

3.1 Creating contents

The first phase is the edition of the educational contents. Teachers must generate the contents related to the target subject. These contents will be as much text as images, audio, animations, questions, exercises, etc.

In many cases, educators create educational courseware which is based on proprietary and not compatible formats. As an XML[1] document contains information about contents and structure, but not about appearance, it facilitates

[1] Extensible Markup Language. http://www.w3c.org/XML/

information reuse, provided it is a valid document. An important feature is guided edition. An XML editor adequate to WLMs must ensure that the generated XML document is a valid document. It would also be desirable that the document were always valid during the creation process in a way transparent to the author. It is important if we do not assume a widespread knowledge of XML by authors. This asumption is realistic if we put emphasis on usability in the university environment, as much in technical as in non-technical areas.

3.2 Splitting contents

The input of this phase will be the XML contents. The output must meet two requirements. The first requirement states that the output must be web-based. The second one states that contents must be divided into several pages. Thus, the output of this phase will be formed by these pages and their structure.

The use of HTML will help us to meet both requirements. First, HTML is the language of the Web par excellence. Second, to divide contents into pages we need to know its visual representation on a computer screen and HTML gives us this representation. This phase is made up of two steps: first, transforming XML into HTML contents, then dividing these contents into several HTML pages and extracting the structure of the contents. Figure 1 illustrates these steps.

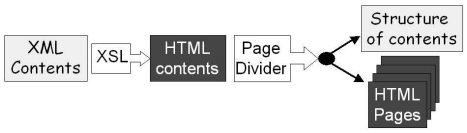

Fig. 1. Transforming XML contents into HTML pages and extracting its structure

To convert XML into HTML contents we use an XSLT. This transformation could be designed by the author or by others depending on the educational context; for instance, a university could have a general XSL for all of its WLMs. The result of this step will be one HTML page.

Now, the HTML contents can be divided into pages. As every HTML element (texts, images, etc.) has known dimensions, we can group these elements into differents pages, keeping the semantic order. Simultaneously, we extract the hierarchical structure of the contents and pages where they are located.

3.3 Browsing contents

In this phase, the interface between the user and the contents will be described. Figure 2 illustrates the process. The structure of the contents and the pages where they are located allow the user to know where he/she is at every moment (orientation information), and where he/she can find other contents (contents browsing) [15].

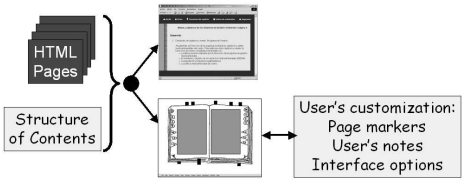

Fig. 2. Two interfaces to access contents

We propose accessing contents in two ways based on the ability to customize interface and contents. A first interface is a non-adaptative WLM, like common websites with navigation aids, menus, tables of contents, etc. The left size of Fig. 3 illustrates this interface. The second interface is a user-adaptative WLM that applies the "book" metaphor allowing the user to add bookmarks and notes, or modify the navigation facilities. The right size of Fig. 3 illustrates this interface.

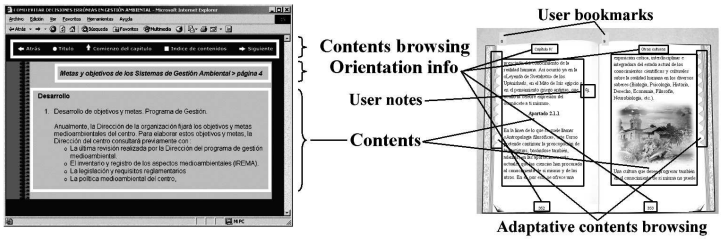

Fig. 3. Adaptative and non-adaptative user interfaces

4 Our Implementation

In this section, the tools developed to implement our approach are described.

4.1 *eXitor*, a XML editor

eXitor is used for the generation of the contents by authors. Its main aim is to guide the user in the process of edition and creation of always valid XML contents. The tool was developed in Java, given its multiplatform and portability features. A screenshot of *eXitor* is showed in Fig. 4.

The development of this tool is summarized in three steps: first, handling the information about the structure of the document[2]; second, it is needed a persistent structure in memory to store the document and to access the contents; and finally, the interface design is split into two parts: a tree view, to maintain

[2] DTDs or XML-Schema in XML terminology.

in a visible way the hierarchical structure, and a text view, to edit the textual content. Further information about *eXitor* is available in [20].

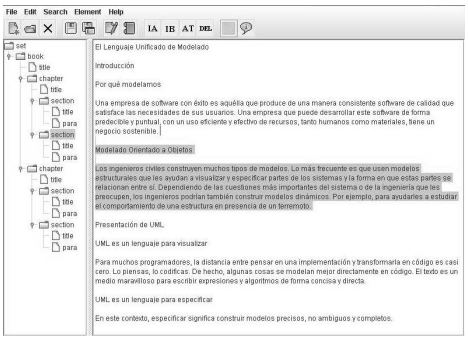

Fig. 4. Generation of XML contents with eXitor

4.2 *JoPee*, a document splitter

JoPee is a tool to generate web-based learning contents, and it needs: a XML document with the contents, a XSL document and information about elements that show the structural levels in the web contents (structural elements).

As the contents' structure can be diverse, *JoPee* needs information about structural elements, which is given by a configuration file (it can also be generated by an interface). The application must check if the structural elements meet the structure. Each level must have only one style format, because HTML tags are used to know when a new level has started. The XSL document contains the correspondence between elements, structural or not, and the style elements.

To divide the HTML document, *JoPee* needs to know the dimensions of the pages to generate. First, a preprocess step must be done, checking dimensions of images; if they are bigger than the dimensions of pages, new scaled images are created and linked with the original ones; checking lists keeping the number of items in ordered lists and the vignettes of the items in unordered lists. The main process checks the dimensions for each HTML element. If they do not fit the current page, a cut point is established and a new page is created. Simultaneously, an XML document is created, which will have the location of the structural elements in the generated pages.

Nowadays, our prototype uses the DocBook DTD[3] to generate courses in the university. In the same way, it uses a particular XSL for DocBook, where the structural elements are: chapter, sect1 and sect2. However, any schema and its corresponding XSL could be used.

[3] http://www.docbook.org

4.3 Browsing contents with *TasKa* and web navigators

With *TasKa* we implement our user-adaptative interface. *TasKa* uses the pages and structure of contents generated by *JoPee* which will be located elsewhere in a web server. Its contents browsing facilities are: access to adjacent items[4] , fixed bookmarks to the table of contents or the beginning of the current chapter, user defined bookmarks, hyperlinks from/to contents and a record of pages viewed. The orientation information available are: page numbers, page headings and thickness of the sides of the eBook. The user can customize the contents by adding notes to pages and customize the interface by adding bookmarks and configuring the browsing of adjacent items (choosing if he only wants to browse chapters, but no sections, and how many of them). *TasKa* can be used with four different resolutions. *JoPee* splits pages according to these resolutions and saves it in the configuration file. If a user wants to change his resolution, pages must be generated again, but reusing the configuration file in addition to the XML contents, keeping the customization of the users.

For non-adaptative interface, any web browser can be used. In this case, orientation information and the contents browsing facilities must be added to the pages generated. Then a HTML template is applied to struct the previous information with contents of the page. Figure 5 illustrates the template application.

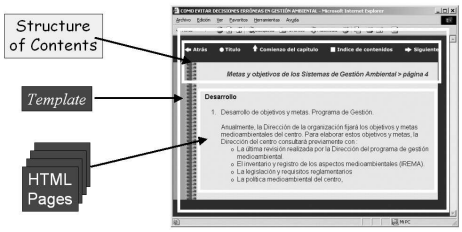

Fig. 5. Non-adaptative material production

5 Conclusions

In this paper, an approach to automatic generation and use of WLMs is described. This approach is a workflow made up of three phases: the contents are created, splitted and browsed.

XML is the language used to create contents, making them independent from its presentation and contents reusing is allowed. A XML editor called *eXitor* is used. It provides guided edition and valid documents generation, so the user do not need a technical knowledge of XML technology. The XML contents are formated and splitted into HTML pages. A tool called *JoPee* is used for it. The

[4] In Docbook these items are pages, sections, chapters, etc.

contents are processed in an automatic way by: formating, sizewise splitting, structure generating and customization reusing. The web contents are browsed in two ways. Using an user-adaptative interface called *TasKa*, which implements the book metaphor; the contents and browse facilities can be customized. Using any web browser as a non-adaptative interface, the contents can be browsed in the same way a web site is used. JAVA is used to develop *eXitor*, *JoPee* and *TasKa* and so, our WLM implementation is platform independent.

Though we are using this approach to build an eBook about java programming, with compiler and visualizations capabilities, we claim that this approach is general enough to produce other e-documentation, not only WLMs, because the creating and splitting phases can work with any structure of the documents.

References

1. A. Arif. Learning from the web: are students ready or not? *Educational Technology & Society*, 4(4):32–38, 2001.
2. C. Chou. Developing hypertext-based learning courseware for computer networks: The macro and micro stages. *IEEE transactions on Education*, 42(1):39–44, 1999.
3. CourseGenie. http://www.coursegenie.com. Last accessed 29/03/2004.
4. F. Davis. User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *J. of Man-Machine Studies*, 38:475–487, 1993.
5. N. Drakos. The latex2html translator, 1999. University of Leeds. UK.
6. EasyGenerator. http://www.easygenerator.com. Last accessed 29/03/2004.
7. EasyProf. http://www.easyprof.com. Last accessed 29/03/2004.
8. C. Kohls et al. Using TCL to design interactive e-learning materials. *The Magazine of USENIX & SAGE*, 28(1), 2003.
9. C.M. Boroni et al. Engaging students with active learning resources: Hypertextbooks for the web. In *SIGCSE 2001*, pages 65–69. ACM Press, 2001.
10. G.S. Akerlind et al. Enhancing self-directed learning through educational technology: When students resist the change. *Innovations Education and Training International*, 36:96–105, 1999.
11. I.Fischer et al. An example of generating internet-based course material. In D. Kalpic and V.H. Dobric, editors, *ITI 2000*, pages 229–234. SRCE - UCC, 2000.
12. J. Sheard et al. Challenges of web-based learning environments: Are we student-centred enuf? In *ICWL 2003. LNCS*, pages 1–11. Springer-Verlag, 2003.
13. K. Baas et al. A practical model for the development of web based interactive courses. In *FIE 2001*, 2001.
14. P. Crescenzi et al. A tool to develop electronic course books based on www technologies, resources and usability criteria. In *ITICSE 2003*, pages 163–167, 2003.
15. P. Diaz et al, editor. *Libros Electrónicos*. RA-MA, 1996.
16. P.O. Au et al. A web-based platform for e-learning based on information management system. In *ICWL 2003. LNCS*, pages 46–54. Springer-Verlag, 2003.
17. R. Martínez-Unanue et al. Electronic books for programming education: A review and future prospects. In *ITICSE 2002*, 2002.
18. R.J. Coffin et al. Motivational influences on computer-related affective states. *Computers in Human Behavior*, 15:609–623, 1999.
19. S.S. Ong et al. Experiences in developing and running WebCMS. In *ICWL 2003. LNCS*, pages 55–68. Springer-Verlag, 2003.
20. V. Fresno-Fernández et al. eXitor: a tool for the assisted edition of XML documents. In *ELPUB 2003*, pages 308–315, 2003.