Design and implementation of a user friendly environment for Learning Objects creation

Giovanni Adorni, Diego Brondo, Mauro Coccoli
University of Genoa, Department of Communication,
Computer, and Systems Science
E-Learning & Knowledge Management Laboratory
Viale Causa, 13 - 16145 Genova, Italy
{adorni, diego.brondo, mauro.coccoli}@unige.it
WWW home page: http://elkm.unige.it

Abstract. The aim of this paper is describing the SCOMaker, a user friendly authoring tool for the automated production of SCOs, mainly oriented to distance learning and learning object design. It is a freely available open source application to assist teachers and academics in the publishing of web content without the need to become proficient in HTML, XML markup or any other language. SCOMaker can export content as self-contained web pages or as SCORM 2004 packages usable on a LMS server or designed for a database SCO repository. This project has been developed by ELKM (E-Learning & Knowledge Management) and led by the University of Genoa..

1 Introduction

The progress of Information and Communication Technologies (ICT) is changing in depth production, management, and exchange of knowledge: methodologies and tools for communication, investigation, and information research are rapidly changing too [1]. Several years of experience in developing e-learning projects at the University of Genoa [2], have pointed out a scenario of increasing managerial complexity, due to the existence of heterogeneous technologies. In an open and heterogeneous environment, the interaction among different tools and platforms, and legacy systems can only be obtained by means of seamless integration and communication, and the use of standards. Another key point is the usability: researchers, teachers, and anyone else having to develop learning objects, has to cope with complex technologies, programming language and elaborate software. Hence the objective of the project is to develop an easy-to-use [3] tool enabling authors of educational materials to produce and deliver sharable objects and documents

(compliant to international standards, ADL SCORM [4] in primis) in a short time and with a little effort. In this respect, a specific tool has been designed, which we called "the SCOMaker". Furthermore the SCOMaker is thought as a part of a comprehensive pipeline of software tools that cover the whole process of course design, from concept map indexing to publication or storage passing through the creation of learning objects [5]. SCOMaker can be adapted to several application scenarios and learning backgrounds (school, university, primary school, business). On account of this and on the different skill proficiency of potential users, it was of primary relevance to develop a system respecting two main concepts: being accessible and easy-to-use.

2   Software architecture

The first step, in building educational materials, is to create a concept map, a logical and well-formed diagram showing the relationships between concepts of the topics of the lesson. This diagram is a good way for encoding knowledge and
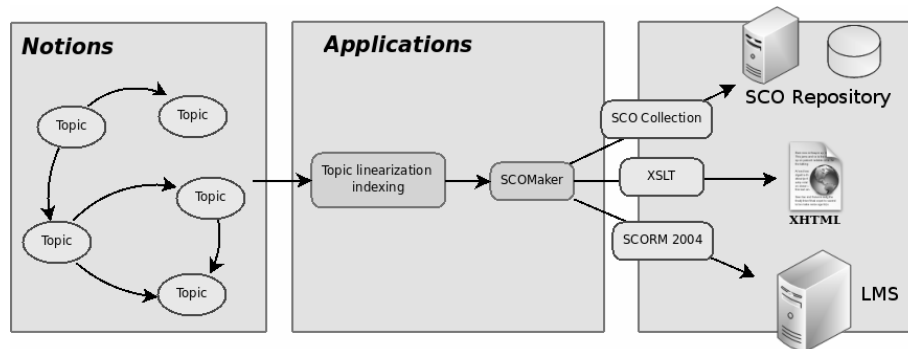

Fig. 1. Learning object production line

connecting this encoded knowledge to relevant information resources. In the notions schema, shown in Figure 1, topics represent the subjects of discourse and arrows are the associations, representing relationships between the subjects.

Following topics association it is possible to glean a linear sequence of topic obtaining a well ordered index (chapters, sections, subsection, etc.). The concept map represents in all respects the index of the e-learning course and, due to its intrinsically hierarchical nature, it can be easily turned into a XML tree following the priority rules. The XML file index can be imported into the SCOMaker creating the framework of the project. The second step is the creation of the content based on the above logical structure and this can be obtained in two different ways. First way writing from the scratch and importing documents and multimedia contents. Second way importing and adding content to an empty framework solution obtained with the topic linearization. Any topic can be linked by author to one or more occurrences, which connect the subjects to pertinent information resources (i.e. PDF documents,

images, multimedia files, etc). Such contents are organized in the XML file and from this point on it can be re-edited or rendered in different shapes. Moreover users can modify the framework freely.

3   Implementation

We developed a web application written in PHP, a reflective programming language, and used XHTML, a markup language that has the same depth of expression as HTML but with a syntax that conforms to XML syntax, for the web interface [6]. This guarantees documents to be well-formed and a better accessibility.

A significant advantage of building web applications to support standard browser features is that they should perform as specified regardless of the operating system or OS version installed on a given client. Additionally, the ability of users to customize many of the display settings of their browser (such as selecting different font sizes, colors, and typefaces) can give an advance to the web application accessibility [7]. SCOMaker was developed taking into account an user-centered design [8], with its intended users in mind at all times. The web interface was designed to be as easy as possible, just like an internet page and all the complex operation for contents and outputs manipulation are completely transparent. So that teachers, instructors or business managers have to focus their attention only on the contents of the lesson .

3.1   Content management

Contents could be of different type: text, images, HTML pages, WiKi, multimedia files imported into SCOMaker with the use of an URI to the remote resource, or embedded ones written directly with the SCOMaker editor.

Each asset (text field, image, multimedia content, etc.) is handled by a specific PHP class. These classes handle both the module interface providing a suitable editor for each content type and the object manipulation methods (PHP functions). This way of project makes possible to developers to easily add new features, fulfilling new educational purposes, just making new modules for handling different type of contents. This modular approach guarantees flexibility and adaptiveness to different pedagogical methodologies and a further extension of SCOMaker features or new standards to come.

Once obtained a structure complying to own intents, the only thing left is the users to save the entire project. Thanks to a set of plugins the customers can choose among different output formats. Using a module rather than another, the learning structure will be converted into different formats for different purposes.

3.2   SCOMaker outputs

Users can create XML output of the course they are creating for a later time modification; or trough a SCORM 2004 standard converter can create a Sharable Content Object portable on any LMS platform SCORM compliant. Moreover they

can send the output to a database with the purpose of adding their work into a learning objects repository. Finally they can create a web version of the lesson in XHTML pages readable on-line through suited XML/XSL-Transformations. This modular approach gives robustness to the standard changes, or database repository changes.

4   Conclusion

Much web publishing applications for authoring such as Frontpage and Dreamweaver provides a set of powerful and complex tools whose use implies a somewhat computer knowledge and a fairly steep learning curve. Moreover too generic software doesn't give appropriate e-learning solutions [9]. With SCOMaker we tried to identify the elements of a learning resource and offer them in an easy and user friendly way. Users in out-and-out learning building, can use SCOMaker features with extreme ease and often without the need for data conversion, thus saving on potential data losses and incompatibilities.

5   Bibliography

1. W. Horton, Evaluating E-learning (ASTD 2001)
2. G. Adorni, M. Coccoli and G. Vercelli, Integrated Management of Courses in a Distance Learning Environment, In Proc. of the Twelfth International Conference on Distributed Multimedia Systems, Bridging the gap between communication and intelligence. vol. 1, pp. 222-227 (2006)
3. Butow, User Interface Design for Mere Mortals (Addison-Wesley 2007)
4. ADL Initiative, SCORM 2004 2nd Edition Overview in ADL (Advanced Distributed Learning) (2004)
5. G. Adorni, M. Coccoli and G. Vercelli, EifFE-L: e-learning Experiences with the Platform of the University of Genova, Journal of E-Learning and knowledge society vol. 1, No. 3 (2005)
6. T. Felke-Morris, Web Development & Design Foundations With XHTML (Addison-Wesley 2007)
7. J. Slatin and S. Rush, Maximum Accessibility: Making Your Web Site More Usable for Everyone (Addison-Wesley 2003)
8. J. Lazar, Web Usability: A User-Centered Design Approach (Addison-Wesley 2006)
9. G. Bonaiuti, E-Learning 2.0 Il futuro dell'apprendimento in rete, tra formale e informale (Erickson 2006)