

USING XML AND XSL-T TO PRODUCE AND MAINTAIN COURSES ON THE WEB

M. Nanard¹, J. Berger¹, K. Bianciotto¹, E. Verdoire¹

The CNAM “Conservatoire National des Arts et Métiers” is an university created in Paris in late XVIII century, at the end of the French revolution. Since its opening, CNAM is explicitly dedicated only to continuous education. Its organization was already quite novative : courses take place only at night so that workers may freely attend without conflicting with their professional activity. More than two hundred years later, the CNAM still is the premier French university concerned with continuous education (<http://www.cnam.fr>).

The usability of internet and the cheapness of home computers is causing a drastic change in continuous education. New technologies make distant learning really usable by anyone now. As a consequence, CNAM has initiated a large program to develop courses on the web, and to make them available to registered students on the CNAM private web portal.

In parallel to the effort to adapt the pedagogy and the course contents to the new technology, several studies and experiments are conducted to improve the efficiency of web courses development. Mastering new technologies for developing higher quality courses on the web at a lower cost obviously is a very important challenge for CNAM.

The experiment

Developing courses on leading edge of technology and on fast evolving domains has some specific constraints : such courses require a yearly maintenance to keep their contents up to date in order to remain valuable and attractive. The development of a course on “HCI (Human Computer Interaction)”, a very fast evolving domain, is a wonderful test-bed for experimenting, improving and mastering new technologies for automated course production and maintenance. The used technology aims at providing reliable solutions to preserve the investment of a course development for the long term. The low cost maintainability of the course is one of the major topics of the experiment.

Background: Designing a document for distant learning is far different from preparing an *ex-cathedra* course or even from writing a book. Since the teacher is not present when the student is learning, she must take care of supplying in digital format all informations and the rhetoric which usually is transmitted by voice and corporal expressions. Beyond rhetoric, mastering consistent typography and interaction techniques is required to produce attractive and valuable e-documents.

The process which turns a course into an e-course cannot be reduced to the simple translation of documents, initially designed for print, into HTML and Java code. E-courses documents have to be designed especially for that purpose. Two categories of people have to cooperate : the author who owns the knowledge and masters the rhetoric to deliver it, and the developer who is aware of the web development technology and of cognitive aspects of typography and interaction. Unfortunately, both skills are rarely present at the top level within a single person. This leads web designers and teachers to collaborate intensively. Thereby, a complex task of coordination between them is required. The structure of the web site is usually designed jointly by teachers and designers, whilst the actual HTML and Java coding

¹ Conservatoire National de Arts et Métiers, France
mnanard@lirmm.fr, berger@cnam.fr, bian@lirmm.fr, verdoire@cnam.fr

is done by the developer. Since all these tasks are performed manually, any late change either in the e-course design in terms of structure, of look, or even of interaction is extremely costly. For instance a deep layout change leads to edit the whole page set. For the same reason, maintenance of the e-document is a heavy budget line if some parts of a course need to be yearly upgraded, such as it occurs in courses about new technologies.

Principles of XML and XSLT approach

The HTML page maintenance problem is not specific to e-learning. It concerns Web technologies as such. Thereby, solutions have been proposed by the W3C, the “World Wide Web Consortium”, as early as 1996. XML, “the Extensible Markup Language”, (<http://w3.org>) and some of its associated languages XSLT, SVG and SMIL, drastically reduce both the production and maintenance costs of web sites. Today, XML is already a well admitted standard. It is fully supported by navigators such as Netscape and IE5. Tools for XML and XSL based production of HTML pages are available on the marketplace.

The underlying principle is quite simple: XML, as its name suggests it, is an extensible language whose purpose is precisely to define target application dedicated languages, for instance a language for describing courses. Such a language can be made as simple as wished and easy to use by novices. Since the language tags can be freely chosen by the teacher to denote the abstractions she handles within the course, she has nothing more to do with the boring HTML tags, and can focus on her own expertise.

Nevertheless, for portability reasons and for usability on old browsers, it still is suitable that the actual web pages be coded into HTML. To do so, we use a second XML based language called XSLT. It is used to describe, once for all, how the teacher defined abstractions, which occur in the XML coding of the course contents, are to be translated into HTML. It releases from the burden of coding each page in HTML. To generate the whole set of pages of the course, an automata takes as input, on the one hand, the translation rules described in XSLT, “the Extensible Stylesheet Language Transformation”, (<http://w3.org>), and, on the other hand, the course source files described in a XML compliant private language. As a consequence the two partners now have a simple and consistent way for interacting : the teacher delivers XML files and the designer delivers XSLT translation rules.

By providing a methodological support which clearly separate the roles, each of the partner can work independently, each with her own expertise, without interference. On the one side, the author may add, at will, any new contents to the course or reorganize it. With respect to the XML based course description language, one has just to run the translator to get the updated set of HTML pages. On the other side, the designer can improve, at will, the design and specify better HTML translation without interacting with the course contents. Each of such changes are automatically reported wherever the associated abstractions are used.

Practical application of XML XSL for the HCI course development.

On the author's side: The previous organization has been enhanced to provide better comfort to the teacher who produces the course contents. Since teachers are more used to edit documents with texts editors such as MS Word rather than with structured XML editors, or even with authoring tools, we have added a preliminary step to enable the author to go on editing the contents with her familiar tools. A simpler tagging language syntax has been defined. Such tags are directly placed within Word documents. A preprocessor automatically converts this language into XML. Thereby, the author has not to be aware of XML syntax,

and preserves her own way of working. For instance, she may, still use MS Word spelling and grammar tools to check the source files before production.

The preprocessor takes advantage of this pass to break down the contents into the targeted Web site pages structure, and to prepare the navigation structure. For instance, it automatically prepares some specific XML code later used to elaborate list of definitions, cross references, indexes, tables of contents, and links from words to their definitions. Since the pre-processor processes the whole set of sources files, any addition is automatically taken into consideration. For instance when the author adds a new definition somewhere in the course, the indexes and cross references are automatically updated with no more effort. For instance, to do it, the author has just to insert a “*\$def*” tag just before the sentence which contains the concerned definition.

The pre-processor makes the markup syntax lighter for the author by suppressing the XML explicit hierarchical structure notation based on “slash tags”, and thereby reserves the source text readability. Starting from simply tagged text also provides a cheap reverse engineering technique for turning paper based courses into well structured and interactive Web courses. It also provides a reverse engineering technique for turning paper based courses into well structured and interactive Web courses.

On the designer’s side: Since the development stage is fully automated, a larger part of the budget can be invested into the design. Instead of developing pages, the designer focus on generic page models. She has just to prepare, once for all, the HTML code for each constituent of the models and to place it into the XSLT templates as translation of the XML structure. Since the designer’s task can be performed without working on the actual content, the production schedule can be tighter, each of the partners works in parallel. Furthermore, experimental feedback can be taken into account at nearly no cost : whenever a flaw, even a minor one, is observed in the design, the designer has just to fix it in the model description. Only a few minutes are necessary to consistently upgrade hundreds of pages in the Web site, and one can be sure that all occurrences are correctly updated since all the pages are re-generated according to the updated models.

In this approach production and maintenance are not distinct. One may easily start evaluating the course and improving its quality far before the production is over, thereby quickly integrating the impact of the growing know-how of the development team into the product. The well know drawback of the classical “specify first then do” approaches, where it is often too late and too costly to fix minor flaws once they become visible, completely disappears with this technique. Taking student feedback into account is quite easy and inexpensive.

Conclusion

Using jointly XML and XSLT enable to clearly separate the course content production from its transformation into a set of well organized and well presented interactive HTML pages. The designer focuses on generic pages models to improve the overall consistency and to organize the navigation structure according to pedagogic choices. The author can fully focus on her expertise do design contents and rhetoric without being hampered by page layout considerations. The automated generation process ensures a reliable target document fully in conformity with the defined models. Using XML and XSL-T to separate contents from presentation makes contents elaboration easier and production simpler, cheaper, faster and the overall more consistent.