A TOOL FOR CREATING UNIFIED COURSE CONTENT

Ivan MADJAROV
IUT d'Aix-en-Provence, Département GTR
Université de la Méditerranée, France
Bogdan SHISHEDJIEV
Technical University of Sofia, Bulgaria

Abstract: In this article, we propose an extensible framework based on XML technologies for the development of an e-Learning hypermedia system accessible via the Web (XESOP). The article insists on the adoption by the users of an e-Learning system by the deployment of semantic tools, which can exchange information in a flexible way. This work consists in searching semantic supports for the design and the storage of the learning objects for an open and distance learning.
Key words: Learning Objects, e-Learning, XML.

1. INTRODUCTION

The development of the Internet emerged new needs for the distribution of information. In the learning area notable changes appeared. One finds more and more courses on-line designed for an open and distance learning. Now Internet is used largely as a learning tool since it offers learning anytime, anywhere. Technology is the key factor in e-Learning and it is used for developing and delivering the learning program.

The objective of presented work is to define a general framework specifying semantic contents for e-Learning including multiple elements in order to propose various formats of presentation. This context requires new specifications adapted for e-Learning, by defining a pedagogical resource as the smallest unit being able to be carried out by a student: an image, a text, a multi-media resource, etc (*Learning Objects*) [1]. Based on a study of various descriptions of metadata for teaching (DTD – *Document Type Definition*) our work proposes a schema, a grammar of validation (XML Schema) of generic type including courses and exercises developed in XML (*eXtensible Markup Language*) [7]. Finally, we propose an original step aiming building and storing courses. This goal can be reached only while being based on standards and by avoiding any limitation due to a platform or a specific operating system.

2. XML IN E-LEARNING

The factors, which need to be addressed for the success of e-Learning are the clarity of the learning objectives, the quality content and the interactive learning techniques applied using multimedia support. Mostly an e-Learning programme will involve development by number of people simultaneously and design and development of standards need to be put in place to ensure consistency and transferability of skills. E-Learning is a teaching process with its own needs which are based on the creation of learning objects and their manipulations: storage, exchange, publication and distribution.

The XML standard allows us to define our own mark-up languages with emphasis on specific tasks, such as e-Learning, data management, and publishing.

For creating a mark-up language, the information about the elements, attributes, and rules for their use is stored inside of a DTD. DTDs were designed to serve the needs of narrative-style documents. XML is being used for object serialization, stock trading, remote procedure calls, vector graphics, and many more things that do not look like traditional narrative documents and it is in these new areas that DTDs are showing some limits. The first limitation is the almost complete lack of data typing, especially for element content. The second problem is that DTDs have unusual non-XML syntax. We actually need separate parsers and APIs (*Application Program Interface*) to

handle DTDs than we do to handle XML documents themselves. The third problem is that DTDs are only marginally extensible and do not scale very well. It is difficult to combine independent DTDs together in a sensible way.

XML Schemas are an attempt to solve all these problems by defining a new XML-based syntax for describing the permissible contents of XML documents that includes: powerful data typing with range checking, namespace-aware validation based on namespace URIs rather than on prefixes, extensibility and scalability.

The development of educational metadata standards as LOM (*Learning Object Metadata*) [1] is a proof about the importance of this approach.

There are two major aspects of authoring learning objects: first, the content of the learning object itself, and second, the metadata describing the learning object [16]. Both are accomplished by XML, which uses tags to structure information and which refers presentation information to a separate document entirely (XSL file) [8]. By organising content in this way, print versions, web versions or even wireless versions may all be produced from the same XML. In any case, it is not reasonable to use one language for all parts of an online course. They are a set of different languages for different parts. Related sets of specifications are being defined by the W3C, such as MathML (*Mathematical Markup Language*) [6] for creation and the presentation of the mathematical expressions, the SVG (*Scalable Vector Graphics*) [5] dedicated for the creation and the presentation of vector graphics. Rather than use a single tool, such as an XML editor, course authors will begin to use, tools (semantic editors) designed for specific purposes in an integrated environment.

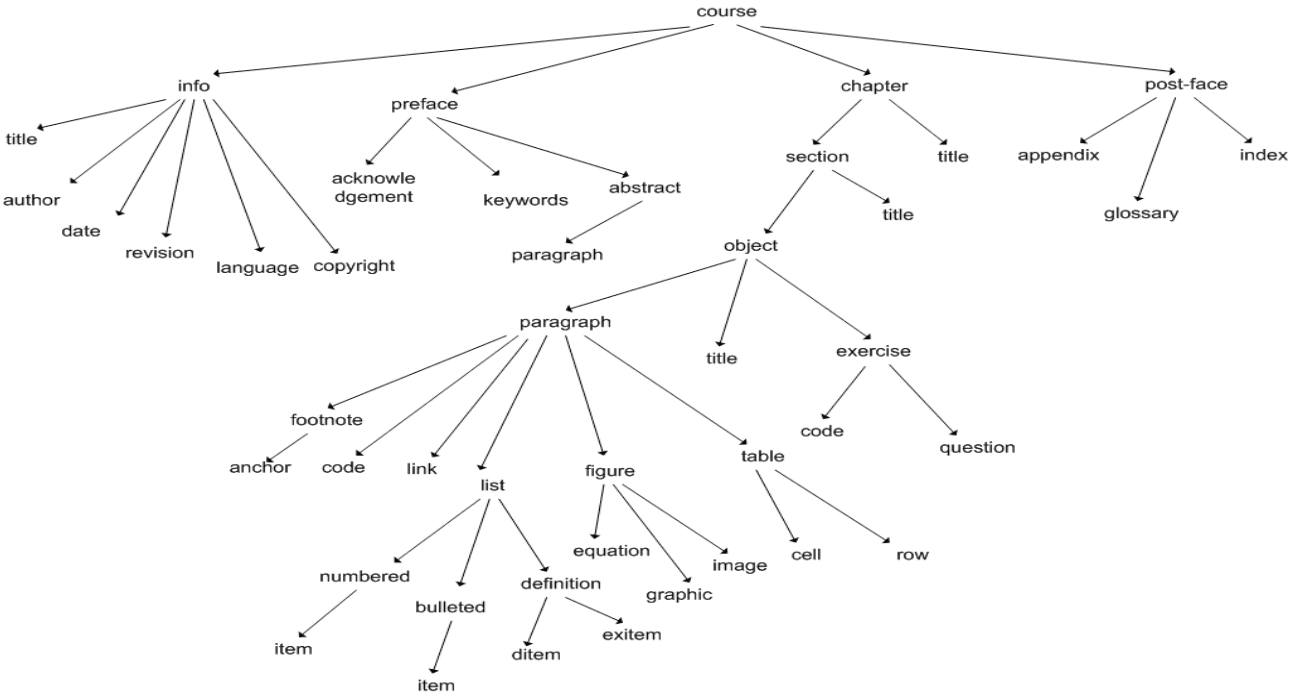

Fig. 1 The structure of the course

There are many implementations of XML DTDs for structuring educational content. After reviewing most of these approaches, we found that they share two common characteristics: they are too complex or lack certain features we wanted to use in our project. For these reasons, we designed an XSD (*XML Schema grammar*) in order to enable us to structure the content of a generic Learning Object and to enrich it with a set of metadata. The root element of the Schema is the *course*. It consists of the elements *info, preface, chapter and post-face*. *Info* is a metadata element conform to LOM [1] and contains common information like the *title*, the *author(s)*, the *language* used, *copyright,* the *date* of last modification and others. The *preface* category contains common

information like the *acknowledgment*, *keywords* and the *abstract*. The *chapter* element consists of *section(s)* with a *title*. A section may contain an object or multiple objects or other sections and thus permitting creating documents complex structured documents. Thus enabling hierarchical structures of the *chapter-section-object* contains non-structural headings or *paragraphs* within which other objects can be used. Mathematical equations can be expressed by the use of MathML and SVG is supported for high-resolution vector graphics. Each of these objects can be decomposed into *code, paragraphs, links, figures, lists, tables, footnotes* etc. Each figure can be composed into *equation, graphic* and *image* (Fig. 1).

The benefit of structured content is the separation of content and style. Content may be reused in other presentational formats. Apart from this obvious advantage, storing the content of a Learning Object as XML data allows for an easy transformation into several output formats, such as HTML or XHTML for online browsing or the PDF (*Portable Document Format*) for offline viewing or printing. These transformations are realized by means of the XSL, who is divided into two parts: the XSLT (*Extensible Stylesheet Language Transformation*) and the XSL-FO (*Extensible Stylesheet Language Formatting Objects*) [17].

3. THE SEMANTIC TOOLS DESIGN

Current XML editors allow publishing any document respecting a grammar. The majority of them are able to validate a XML document while being based on a grammar defined according to a description of the type DTD. They are still rare the editors respecting at the same time a grammar DTD and a grammar XML Schema. The most advanced editors propose complex functionalities, which prove to be difficult to understand by an end-user.

A XML semantic tool for learning objects edition must carry out functionalities defined within the framework of a specific problem. It must handle the semantic contents of a XML document through a grammar defined for a particular case and must be able to treat the corresponding semantic contents in a general framework for the construction of course. The realization of such concept leaves the framework of a general XML editor.

For the creation and the edition, the learning objects our study led us to develop a semantic tool (editor) of the type WYSIWYM (*What You See Is What You Mean*). To meet the needs for e-Learning [2], we retained the languages SVG [5] and MathML [6], based on XML, and which are used respectively for the description of the scalable vector graphics and the mathematical equations. We created two editors based on these languages by attaching them to a basic editor in the form of *plug-ins*. This concept ensures a flexibility to add new functionalities to the basic author system in the future.

With regard to the distribution of the courses and their presentation we chose the style sheets XSL(T) [8], a meta-language which, starting from the contents of only one XML document can produce several types of presentations: type slide (HTML, XHTML), Web pages, hard copy (PDF, RTF) or other.

For a presentation, the author defines the contents by selection for each type of presentation. For the creation of the various presentations, we propose a simple principle: the selected parts of contained are marked by adding owner's tags; they indicate to the tools of transformation the sequences to be extracted. The pre-presentations are saved in a native XML database and can be modified later on. The final presentations are created after choice of a format of transformation (HTML, XHTML, PDF, RTF, etc). All the modifications are done starting from the original documents.

Learning objects of type vector graphics (SVG) and mathematical equations (MathML) can also belong to a course. Objects of the multimedia type (sound and video) are stored in the learning objects with their links (URI). Their animation is possible starting from SMIL [11] *plug-ins*. The semantic editor can insert external images according to the standards adopted on the Web (GIF, JPG, PNG).

Figure 2 illustrates the functionalities of our system. For the same document containing a course we can create various kind of issues using a different XSLT processor. Certain elements of the document can be omitted; other will not appear in the same order as in the document of origin. Initially, the document source is filtered and reordered, then, it is possible to apply a new structure to him.

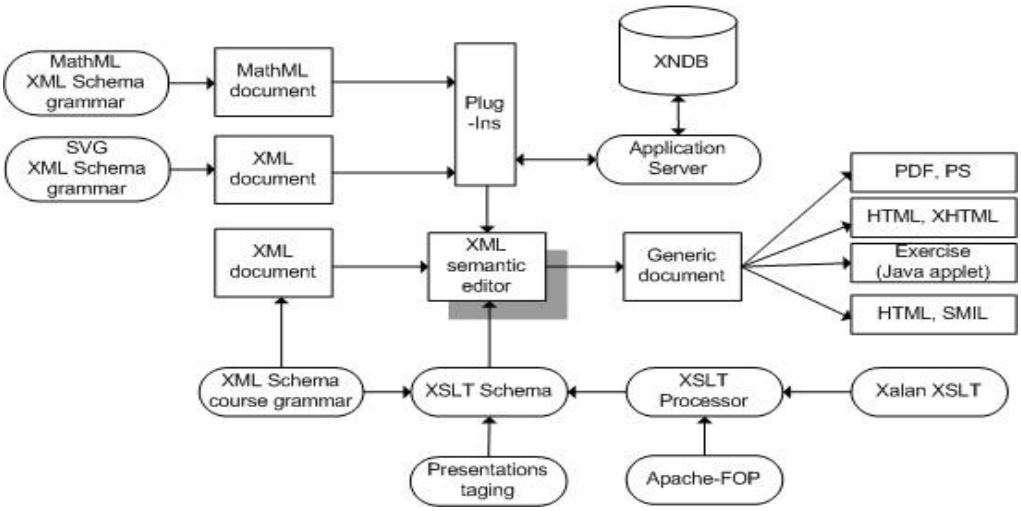

Fig. 2: Functional diagram of the semantic editor

The handlings of an educational XML document are the reading and the modification of the contents. This handling is done using two different approaches: SAX and DOM. The choice between SAX (*Simple API for XML*) [12] and DOM (*Document Object Model*) [13] depends on the objectives of treatment. The practice shows clearly that SAX is faster and easy to use. On the other hand, DOM is adapted for the management of documents where the tree structure has an importance. A tool as JAXP [15] (Java parser of the type SAX) replaces neither SAX nor DOM, but only makes it possible to encapsulate the implementation of the parser. JDOM [15] (java tool for the development of DOM applications) makes it possible to work with SAX and DOM and seems to be a good solution, because this tool is optimized for Java. For these reasons, JDOM is the parser than we used for the treatment of XML documents in our system.

MathML [6] is a language, based on XML, which allows the creation of *mathematical expressions* whose syntax and grammar are specified in a XML Schema. For the design of the WYSIWYG mathematical plug-in editor, we used APIs of package *Jeuclid* [3]. It is a whole of classes, which implement the visualization of MathML documents on an object of the class `java.awt.Graphics`. This component converts documents of the type MathML (`mml`) into GIF or SVG images.

The SVG language was proposed by W3C like a language of description of 2D graphics in XML for the realization of the scalable images [5]. SVG allows the use of three types of graphic objects: vector forms (lines, curves), images (PNG, JPEG, SVG) and text. SVG graphics are interactive and dynamic. Their animation can be defined either inside SVG files or in an external language (SMIL for example). For the design of SVG editor, we used a collection of *open source* tools written in Java belonging to the project *Batik* (Apache) [12]. It uses all XML tools like parsers, transformers, and databases. Conformity with the namespace makes it possible to mix XML grammars between them. Thus, a XML document can contain SVG graphics, mathematical expressions in MathML, and presentations in SMIL.

The concept of the various presentations is based on the choice of various parts of the same contents retained by the semantic editor. Any presentation starting from a XML document must undergo a XSLT transformation by the application of a suitable style sheets. To avoid the multiplication of style sheets for each presentation, a practical possibility is to add specific tags in

the source XML document. Thereafter, these specific tags are not taken into account by the parsers of the selected formats. For the export of the presentations in the formats of paper quality such as PDF, PCL, PS or others, we profited from certain results of the FOP (*Formatting Objects Processor*) project [13] of Apache. The XSL-FO documents are style sheets, therefore XML documents, with a name space a little special, which defines the attributes necessary for page setting. These XSL-FO documents are created by a XSL transformation of a XML document.

4. CONCLUSION

The presented work in this article relates to the design and the development of *a XML semantic editor for learning objects*. This editor is integrated in the platform of e-Learning - XESOP. The tests carried out these last months showed the weak points of the realization, but also showed the strong points of the design. With the gleam of these first experiments, we can release the following prospects:

- The development of an XSL editor so that the author personalizes each presentation appears a significant element. This editor in the form of one *plug-in*, whose current editor remains open, will bring much flexibility to the final presentations.
- New *plug-in* treating the SMIL tags for the multimedia sequences would be interesting, considering the possibilities of presentation of the courses that could bring.
- The development of the functionalities of the semantic editor could bring us to the design of one *plug-in* treating the questionnaires. This type of presentation of the learning objects closely related to the evaluation of knowledge is largely widespread in several scientific matters. A solution could be largely supported by the existing standards in this field.

The study that we carried out, the tools and the architecture, which we developed, can be used as a basis for later studies and developments since this work remains open in its design. The addition of new functionalities will increase the performances and the possibilities of the system XESOP that we conceived.

5. REFERENCES

1. IEEE, *Learning Object Metadata*, http://ltsc.ieee.org/wg12/, 2003.
2. Nilsson M., Palmer M., Naeve A., *Semantic Web Meta-data for e-Learning, Some Architectural Guidelines*, Proceedings of the 11th World Wide Web Conference, Hawaii, 2002.
3. *Jeuclid Project*. http://jeuclid.sourceforge.net/.
4. W3C, *XML Schema*. http://www.w3c.org/XML/Schema.
5. W3C, *SVG Specification*. http://www.w3.org/Graphics/SVG.
6. W3C, *MathML Specification*.http://www.w3.org/Math/.
7. W3C, *Extensible Markup Language (XML) 1.0*, 2000. *http://www.w3.org/TR/REC-xml*
8. W3C. Extensible Stylesheet Language (XSLT). 2000. http://www.w3c.org/style/xsl.
9. Apache, *Project Apache Batik*. *http://xml.apache.org/batik*
10. Apache, *Project Apache FOP*. *http://xml.apache.org/fop*
11. W3C**,** *Synchronized Multimedia Integration Language (SMIL) http://www.w3.org/TR/smil20*.
12. SAX. *XML parser*. *http://www.saxproject.org*.
13. *Document Object Model (DOM)*. *http://www.w3.org/DOM/*.
14. JDOM. *Des API Java pour travailler avec XML*. *http://jdom.org*.
15. *Java API for XML Processing (JAXP)*. *http://java.sun.com/xml/jaxp/*.
16. Downes St. *Learning Objects*. University of Alberta. 2000.
17. Michael Kay, *XLST Programming Reference,* Wrox Press Ltd.,2000