An Extensible Framework for Building Interactive Courses on Web

Sabin Corneliu Buraga
Faculty of Computer Science, "A.I.Cuza" University of Iasi
Berthelot Str., 16 Iasi 6600 Romania
Tel.: +(40) 32 201529, e-mail: busaco@infoiasi.ro

Abstract

The World-Wide Web, the world's largest hypertext structure, has already in the last ten years a huge development. Recently, many worldwide consistent or semi-consistent collections of scientific data, in specific disciplines, have become accessible on Internet. These sources of information comply with a standard for interoperability. In this paper we propose an extensible framework based on XML (Extensible Markup Language) family for building interactive courses on Web. This framework consists in several parts used for distributed storage of hypermedia, for synchronized multimedia presentations, and for authorization of users (professors and students) to access data. Therefore, increased usage of online courses by non-specialists has increased the need for a more effective and friendlier user-interface experience. One solution of these problems is the use of extensible and adaptable standardized methodologies and languages for annotation of hypermedia information on Web, such as XML (Extensible Markup Language), RDF (Resource Description Framework) or SMIL (Synchronized Multimedia Integration Language). Our proposed framework is platform independent and can be easily implemented on any operating system.

1. INTRODUCTION

The World-Wide Web, the world's largest hypertext structure, has already in the last ten years a huge expansion. Recently, many worldwide consistent or semi-consistent collections of scientific data, in specific disciplines, have become available on Internet. Once simply a means of accessing information stored across various platforms, the Web is now a widely used medium for communication. The Web space has attracted a great deal of attention as a vehicle for delivering distance courseware. Institutions with long-standing involvement in distance education, such as the *Open University* and *Athabasca University* are incorporating Web-based elements.

Although Web-based course materials have advantages over conventional textbooks and lecture notes, they have a number of common deficiencies:

a) Access to course materials is (relatively) slow;
b) Courseware does not adapt automatically to individual students;
c) Interactivity must be programmed (using Java or other programming languages);
d) Features of Web processing (caching and client-side information hiding) interfere with collection of student performance data.

A number of attempts have been made to hit some of these problems, but solutions to one problem often obstruct solution of the remaining problems. *WebWhacker* or *TelePort* applications address the problem of slow access to course materials, by copying Web pages to the client machine (or to a local server), but interactive pages that appeal to various scripts on the Web server do not always function correctly when copied. *InterBook* supports adaptivity and authoring, but all adaptation and page generation takes place at the central server, risking access delays. *QuestWriter* supports authoring, and has built-in client-side and server-side interactivity, but does not adapt presentations to individual students. Another examples are *DreamTeam* and *Dolphin*.

Another issue is the *adaptive annotation* for Web. History-based adaptive annotation is familiar to Web users because any browser allows them to distinguish visited and unvisited nodes, showing these nodes in different colours. The courseware system may offer more advanced methods of adaptive annotation that could be also very helpful for Web users. All adaptive navigation support methods can be based on the models of hypertext theory.

Instead of centralizing processing at a central server, a courseware system must adopt a *decentralized approach* in which overall course management is performed centrally but course materials (text and hypertext documents, multimedia documents, scripts and other applications) are served up locally using software that runs on the student's computer. A co-operative application should make as few assumptions as possible about the system platform with regard to hardware requirements (system performance, screen resolution, multimedia capabilities, other supplementary devices etc.) and software requirements (operating system, version, installed software etc.). Personal computers tend to be unstable and a co-operative application should be tolerant against local breakdowns and should without difficulty allow restarting a system and rejoining an existing group of students.

Interactivity and intelligent tutoring capabilities (e.g. help facilities) must be provided by client-side software as well. Therefore, enlarged usage of online courses by non-specialists has increased the need for a *more effective and friendlier user-interface experience*. Especially for remote students it should be as easy as possible to set up and run a co-operative application. Different applications should provide consistent interfaces, at least for all aspects of installation and general collaboration.

These sources of information and the interactions between professors and students comply with a standard for interoperability. One solution of these problems is the use of extensible and adaptable standardized methodologies and languages for annotation of hypermedia information on Web.

2. EXTENSIBLE MARKUP LANGUAGE (XML)

In order to build a complex distributed system to support interactive courses on Web, it is necessary to understand why this is not possible with classical technologies. Especially the HTML (HyperText Markup Language) standard – *lingua franca* of the World-Wide Web – hinders the development of new applications on the Internet, as it was not designed to do anything else that presenting documents in a Web browser. Documents need to be displayed, processed, rearranged, stored, exhanged, encrypted and so on, in a simple and easy to use manner. It is difficult to express the hierarchical relantionship of data values (course texts, student profiles, professor profiles, marks and other useful information).

For these purposes and for many others, we can use a subset of SGML (Standard Generalized Markup Language), called **Extensible Markup Language**. The *XML* (*Extensible*

Markup Language) language is a recommendation of the World Wide Web Consortium for a meta-language to define markups (annotations) for content publishing on the Web and other areas. The ambition of XML is to provide some benefits not available in HTML, such as arbitrary extensions of a document elements (tags) and their attributes, support for documents with complex structure, and validation of document structure with respect to an elective document-structure grammar, called a *DTD* (*Document Type Definition*). A DTD specifies what elements may occur and their order of occurrence and how the elements may nest in an XML document that conforms to this DTD. Also, instead of DTD, we can use an object-oriented method for validation of XML documents called XML Schema.

Since 1998, XML has grown into a large family of standards integrating key technologies from three previously independent domains: documents, databases, and the Internet. Several examples are *SMIL* (*Synchronized Multimedia Integration Language*), *MathML*, *EFDL* (*Extensible Forms Description Language*) or *RDF* (*Resource Description Framework*).

Using XML, the semantics and the structure of the data exchanged by various Web applications is preserved. The data can be organized as in an object-oriented database. As XML is format-independent it is possible to generate multiple outputs very easily. The XML can be the optimal solution for data integration from multiple sources. The XML documents can be processed either in the Web browser (last generation browsers – Microsoft Internet Explorer 5 and Nescape Navigator 6 – support XML) or on a Web server (e.g. Apache or Internet Information Server).

On the server, the content can be stored in XML documents wihout the layout. Content contributors (course authors, student annotators, different other persons) can write their content in specialized applications that suit their needs and create XML as output. These documents are then stored automatically in a database and can be accessed instantly over the Internet. Using XML on client-side, it is possible to transform XML to HTML in the browser. These transformations can be performed in an extensible way by XSL (Extensible Style Language) stylesheet files. Similar to the Cascading Style Sheets (CSS), the XSL documents separate the content from representation. An XSL stylesheet specifies the formatting characteristics of XML documents on the Web.

For XML documents processing, the Web Consortium was proposed an object-oriented model: *DOM* (*Document Object Model*). A flexible and easy to use DOM implementation is *SAX* (*Simple API for XML*), written in C++ and Java languages. We can use a free implementation of SAX, the *libxml* library, available on Linux platforms, part of *GNOME* (*GNU Network Object Model Environment*) project.

3. RESOURCE DESCRIPTION FRAMEWORK

Resource Description Framework (**RDF**) is a standardized foundation for processing metadata. RDF consists in a model for representing named properties and property values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources and a RDF model can therefore resemble an entity-relationship diagram. In object-oriented design terminology, resources correspond to objects and properties correspond to instance variables. Also, RDF supports the reusability of metadata definitions. The concrete RDF syntax is based on Extensible Markup Language.

The RDF constructs may be used to specify the relationship between various components of the courseware system and between the participants of the online courses. Also, RDF metadata can be used to manage session information:

- Host and user profiles (professors, students and others);
- User profiles of participants of previous and current sessions;
- Session profiles (timing, available resources, etc.);
- Private user applications (editors, viewers, archivers, etc.).

In the following example, we specify in RDF the list of courses accessed by a particular student:

```
<rdf:RDF>
<rdf:Bag ID="courses">
  <rdf:li resource="http://www.infoiasi.ro/courses/web" />
  <rdf:li resource="http://www.es.uaic.ro/courses/management" />
</rdf:Bag>
<rdf:Description about="#courses">
   <User type=”student”>
     <rdf:Description about="http://students.infoiasi.ro/~stud">
       <Name> ... </Name>
       <Year> 3 </Year>
     </rdf:Description>
</rdf:Description>
</rdf:RDF>
```

4. SYNCHRONIZED MULTIMEDIA INTEGRATION LANGUAGE (SMIL)

Based on XML, the **Synchronized Multimedia Integration Language** (**SMIL**) has been created by the World-Wide Web Consortium and is a powerful way to synchronize any type of media (i.e. audio, video, text and graphics) and build time-based, streaming multimedia presentations without the need of learning a complex programming language. The authoring process of interactive online courses built in SMIL is based on the hypermedia model and the courses and presentations can be interconnected.

The SMIL documents can be used to give lecturers to a student group, in a simple and flexible manner. Each page of the online course can be synchronized with the professor’s view. Using complementary methods, the multimedia presentations can include annotations or different summaries.

5. CONCLUSIONS

Using these technologies, we intend to build an extensible framework for building interactive courses on Web. The system components can have a de-centralised structure and the relations between these components will be modeled by RDF constructs.

The courses and all information about the users (professors, students, administrators, maintainers etc.) can be stored in XML documents and can be easily interchanged between different heterogenous computers. The XML technology can be used for course annotations, too. Entire storage architecture of the system will be platform independent.

Interactive hypermedia presentations can be written by using SMIL language. The user interface can be designed in SMIL or in HTML, using XML transformations via XSL stylesheets and the user only need a Web browser to view or to modify the online course.

Our proposed framework is platform independent and can be easily implemented on any operating system, using free software.

REFERENCES

1. Amor, D., *The E-Business (R)evolution*, Prentice-Hall, 1999
2. Berners-Lee, T., *Weaving the Web,* Cambridge Massachusetts, 1999
3. Berners-Lee, T., Connolly, D., *Web Architecture: Extensible Languages*, W3C Note, feb.1998
4. Buraga, S.C., *A RDF Description of Distributed File Systems*, Scientific Annals of "A.I.Cuza" University, Computer Science section, vol.IX, Iasi, 2000
5. Connolly, D., *HyperRDF: Using XHTML Authoring Tools with XSLT to produce RDF Schemas*, W3C Note, Boston, iul.2000: http://www.w3.org/2000/07/hs78/
6. Decker, S. et al., *Knowledge Representation on the Web*, International Conference on Digital Libraries (DL'2000) Proceedings, 2000
7. De Bra, P., *Teaching hypertext and hypermedia through the Web*, WebNet96 - World Conference of the Web Society Proceedings, AACE, 1996
8. Jackson, M., *Assessing the Structure of Communication on the World Wide Web*, JCMC 3 (1), June 1997
9. Myers, D., *A Brief History of Human Computer Interaction Technology*, Tech. Report, Carnegie Mellon Univ, March 1998
10. Roth, J., *How to write shared applications with „DreamTeam“*, Internal Technical Report, Fernuniversität, Hagen, Jan. 1998
11. Streitz, N.A., & Geißler, J., & Haake, J.M., & Hol J., *DOLPHIN: Integrated Meeting Support across LiveBoards, Local and Remote Desktop Environments*, Proceedings of the ACM '94 Conference on Computer Supported Cooperative Work (CSCW '94), Chapel Hill, North Carolina, Oct. 22-26, 1994
12. Todoroi, D., Nazem, S., Jucan, T., Micusha, D., *Transition to a Full Information Society: Stage Development*, Working Paper No. 98-2, College of Business Administration, Omaha, Nebraska, USA, March 1998
13. Welsh, N., *A Technical Introduction to XML*, ArborText Inc., 1998
14. Vassileva, J., *A task-centered approach for user modeling in a hypermedia office documentation system*, User Modeling and User-Adapted Interaction, 6 (2-3)/1996
15. * * *, *World Wide Web Consortium's Technical Reports*, Boston, January 2001: http://www.w3.org/TR