An Investigation of XML-technologies for Infrastructures for Web-based Virtual Courses

Claus Pahl
Dublin City University
School of Computer Applications
Dublin 9, IRELAND

cpahl@compapp.dcu.ie

ABSTRACT

The emergence of XML has lead to a change in the development style for Web sites. XML's advantages, such as the separation of content and presentation, or the possibility to develop application specification markup languages, have changed the creation, management and presentation of Web sites. We will investigate the suitability of XML technologies for authoring and delivering virtual, i.e. Web-based, courses. We will include various educational services including lectures, tutorials, and lab sessions in our investigation. In particular, we will consider the development of content, navigation, and interactive elements for virtual courses.

Keywords

Web-based courses, XML-technologies, e-content authoring and delivery, navigation, interactive services.

1. INTRODUCTION

The creation and delivery of educational systems using the World Wide Web has seen a lot of support over the last couple of years. The emergence of XML-technologies [6] offers a new potential infrastructure for these systems. Since educational Web-sites differ from commercial ones in their objectives, an investigation into technologies for this context is necessary. We shall discuss the benefits and limitations of XML for authoring and delivering virtual courses, i.e., courses delivered using the Web [2,4]. We shall investigate whether the technologies needed to satisfy implementation requirements of advanced courseware do exist.

Our analysis includes the aspects of authoring and delivery. We do not consider collaboration and communication issues here (even though they are of major importance in courseware systems). Instead, we focus on individualized systems. Maintenance problems caused by evolving systems or systems that have to be configured in various ways shall be considered.

We shall review requirements for Web-based educational systems and how they can be matched by suitable XML-technologies. We shall structure our investigation along the activities of authoring and delivery. We start with a brief introduction to virtual courses, and then look at authoring of static content, authoring of dependencies and navigation structures, managing content and finally at the problems arising in dealing with interactive course material.

2. VIRTUAL COURSES

Virtual courses are software environments that provide educational services supporting various learning activities. We shall in particular look at virtual courses that provide a variety of integrated, possibly interactive services ([3] and [5] are examples). Different services provide different views on the material (, or different learning activities around the given material), but their content and presentation shall nevertheless be integrated. Interactivity is an essential part of learning-oriented systems – students learn by doing. This can include simple interactive features such as online multiple-choice questions or more complex ones such as Web-interfaces to simulation tools or other software packages.

We shall look at these courses from two perspectives: a teacher's view focusing on educational or authoring problems and a support view focusing on technical delivery issues.

3. AUTHORING CONTENT

We will look at the authoring of static course material first, i.e., material that is delivered without student interaction. Authoring is the process of generating knowledge and the representation of structured knowledge. Support for authoring should provide at least the following elements: the possibility to define a vocabulary and rules to express knowledge structure and a methodology guiding the authoring process. XML satisfies these requirements in that tags form the vocabulary, DTD-element definitions describe rules how to use these tags, and the document hierarchy lends itself as a guideline for top-down development.

A particular problem is the need of different structural dimensions. We might like to structure the course material in terms of educational elements such as concepts, examples, exercises, definitions etc, but we might also want to use subject-specific tags, e.g., programming concepts (function, variable, assignment) in a computer programming course. The requirement is a namespace technique that allows a course developer to define vocabularies separately and combine them as needed. Modularity is particularly important for the maintenance of evolving systems. XML Schema provides an alternative to DTDs and namespaces offering the required modularity, but widespread tool support is currently not available for schemas.

4. AUTHORING DEPENDENCIES

Dependencies between course elements form a course topology in the virtual world. In traditional classroom teaching these dependencies have to be transformed into a sequence; in virtual courses a more flexible hypertext topology can be realized through which the student can navigate. A course designer needs to be supported by a notation for specifying this dependency structure. The notation should be based on a suitable formal model (e.g., Petri-nets, or StateCharts) for analysis purposes and should allow course developers to describe dependencies between units of the course material in an abstract way. A dependency specification forms then the basis for the creation of the navigation infrastructure.

XLink/XPointer is a technology that allows a course developer to describe links. This includes concepts such as links with a number of targets and also the possibility to annotate links with meta-data. The authoring of individual course units and their dependencies and configuration for the system's navigation structure need to be kept separate. This supports modular design and, therefore, the maintainability of the system. Flexibility is required in order to deal with evolving content and also in order to configure material in various ways for different users or user groups.

5. MANAGING STATIC CONTENT

Activities that shall be subsumed under the term 'management' are editing, parsing, transformation, storage, and retrieval. We do not look at the first three activities here, since they are classical tasks and well supported by XML-technologies (e.g. through XSLT and supporting tools).

The storage and retrieval functionalities are part of both an authoring and a delivery subsystem. In order to support consistency, flexible access and configuration in particular in evolving systems, database support in form of general-purpose systems or XML-databases is needed. The advantages of using XML-databases for storing and retrieving XML-documents are obvious. In general-purpose database management systems two strategies can be followed:

1. Use any existing XML-support (available in some recent systems) or any kind of support for document storage and retrieval (e.g., the SQL99 standard supports large character objects).
2. XML-documents form hierarchies, or trees, whose nodes can be stored as records allowing very flexible access to and configuration of material.

Retrieval is certainly supported by database facilities, but the XML Query language is without doubt a more appropriate tool, since queries are based on the XML elements also used in the creation of the documents.

6. AUTHORING AND MANAGING INTERACTIVE MATERIAL

Content delivery, or content publishing, can in general be supported by XML-technologies such as XSLT or programming language APIs such as Java SAX, which can be used to transform XML-documents into an output format. The specification and delivery of interactive material poses a more severe problem. We assume that a separate tool exists which performs the interaction based on some student input. This tool has to be integrated into the delivery subsystem and interfaced via the Web to the student. More precisely, the processing of XML-documents and external applications has to be integrated. Two alternative scenarios shall be discussed:

1. The XML-document processor, e.g., a Java program using the SAX-API, invokes the external processor, which could return its results in terms of XML-fragments that can be inserted into the XML-document. The output document is then created from the completed XML-input document.
2. The XML-document is extended based on a notation that can be processed in a suitable way. XSP - eXtensible Server Pages – is such an extension of XML [1]. XSP overcomes some JSP (Java Server Pages) drawbacks such as the lack of separation of content and presentation or the output-orientedness. An XSP-document could contain instructions how to invoke an external processor.

We can see from this discussion that an XML-extension towards a distributed computation protocol is needed allowing communication between the server and various applications in an educational system to happen. CORBA-IDL style invocations described in XML-documents would support the solution of this problem.

7. CONCLUSIONS

This investigation of XML-technologies for advanced virtual courses has been motivated by the development of our own virtual course system [5]. The most appropriate existing support framework for our requirements is the Cocoon system from the Apache project [1]. Cocoon is an XML-based Web-publishing framework. Due to the lack of interactivity and database support we opted to develop our system from scratch.

XML is in principle suitable for individualized, integrated and interactive courseware. Competent Web users will be able to create static content and to publish/deliver it. In particular the integration and processing of interactive elements is currently not sufficiently supported and can currently only be implemented involving some software development. Current and future developments for XML Schema, XML Query, XProtocol and XSP are expected to improve this aspect of advanced educational systems.

8. REFERENCES

[1] The Apache Project. Cocoon. http://xml.apache.org/cocoon. 2001.

[2] G. Fox. Uses of XML in Distance and Distributed Education. Workshop on Symbolic Notations on the Web. Tallahassee, Florida, USA, 1999.

[3] T. Inoue and H. Ueno. NACSIS-ILL WWW-Based Training Course. SAINT Workshop on Internet-Supported Education. IEEE Press. 2001

[4] R. Koper. Educational Innovation and XML – the Educational Markup Language EML (in Dutch). SGML/XML Congress. Netherlands Open University. http://www.edubox.nl/artikelen.htm.

[5] C. Pahl. CA309 Databases Course Web-site. http://www.compapp.dcu.ie/~cpahl/teach/ca309.htm

[6] W3C – World Wide Web Consortium. Extensible Markup Language (XML). http://www.w3c.org/XML. 2001