Experience with an XML-Based Syllabus Editor and Search Engine

Michael Wollowski

Introduction

The *Extensible Mark-up Language (XML)* is a promising new technology for creating, maintaining, and searching course pages. XML shows the most promise when it comes to searching for information. Furthermore, the use of XML reduces the overhead of editing web-pages by separating form and contents. Content providers have to do just that, provide the contents. Similar web-pages are rendered by a common stylesheet, which has to be edited just once. A common stylesheet ensures uniform appearance of web-pages, which in turn provides for ease of navigation and recognition of location.

When it comes to course web-sites, there are largely two alternatives: to edit and maintain them directly or to use courseware products such as (WebCT 2002) or (Blackboard 2002). The benefits of direct editing are complete control over appearance and contents; the major drawback is a lot of work formatting contents. The benefits of courseware products are ease of use and integration with other academic systems, such as the Banner; a drawback is that formatting and functionality are as provided. We propose a third option which is slowly become a viable alternative: the use of XML. We will show in this paper that XML shares most of the benefits of the two current alternatives: ease of contents creation and maintenance as well as complete control over appearance and contents. A major benefit of XML is the ability to furnish very precise search engines, enabling users such as students, prospective students, or prospective employers to quickly locate relevant information.

Industry started using XML a while ago, chiefly to unify information between companies and their suppliers (Goldfarb & Prescod 2000). We show an application of this technology that demonstrates its benefits for academic use. We briefly explain the main components of XML, in particular how XML separates form and contents. We will show a web-based editor for XML pages as well as a search engine tailored to a class of XML documents. Finally, we report on some initial user testing of our system. For a more detailed description of the technology, we refer the reader to (Wollowski 2002).

A Brief Overview of the Technology

XML is three technologies in one. It consists of *XML* proper, which is a language for labeling contents. There are *Document Type Definitions (DTDs)*, which are used to specify kinds of documents. Among others, DTDs specify the labels to be used in XML documents. Finally, there are languages to render XML documents. Typically, web-authors use either the *Extensible Stylesheet Language (XSL)* or *Cascading Stylesheets (CSS)*. Since XSL is more powerful when it comes to the way final documents can be composed, we chose XSL over CSS. At this point, only Internet Explorer 5.5 and higher support XSL. Most other browsers support only CSS.

Figure 1 shows an XML file containing course description information for CSSE 100, one of our courses. We show this document rather than the syllabus, because it is shorter. XML *tags* are identified by opening and closing angle brackets. A closing tag is identified by a forward slash preceding its name. An opening and closing tag form an *element*. To aid in readability, we boldface the tags in this write-up.

```
<course_description>
      <id>CSSE 100</id>
      <title>Introduction to Programming and Problem Solving</title>
      <credits>2</credits>
      <description>An introduction to general methods of problem
      solving, structured algorithm design, object-oriented techniques,
      and elementary computer programming.</description>
</course_description>
```

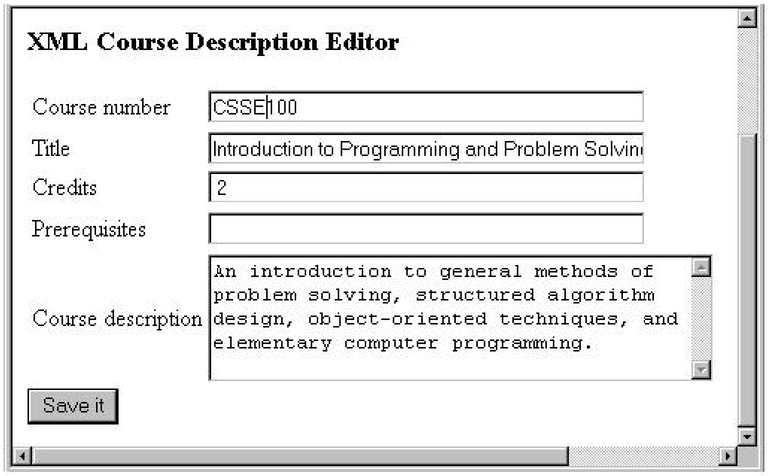
Figure 1: XML document for a sample course description

A key principle of XML is that element names are used to indicate kinds of contents. Notice that there is no formatting information present. This is in contrast to HTML, where tags are used to format contents. This is the primary difference between XML and HTML. We will now explore same of the ramifications of this difference.

Ease of Editing

While content providers may edit XML documents with a text editor, one of the benefits of XML is that one can provide good forms-based editors. In this section we describe such an editor.

A forms-based editor is a web-page which displays text-boxes that correspond to elements in the XML document. The user places information into the textboxes, which will be saved in the corresponding XML elements. Figure 2 contains a screenshot of the editor for our course description pages. When using this editor, authors do not have to memorize element names. We set up the editor so that certain elements come with default information which can either be accepted or edited. An example of such default information is the prefix for course numbers.



XML Course Description Editor

Course number: CSSE100

Title: Introduction to Programming and Problem Solvin

Credits: 2

Prerequisites:

Course description: An introduction to general methods of problem solving, structured algorithm design, object-oriented techniques, and elementary computer programming.

Figure 2: Interface to editor for course description pages

Since XML documents are only concerned about contents, it is very straight-forward to provide a text-field for each XML element. This is in stark contrast to HTML, in which there are no specifications for contents. Naturally, one can use the same editor to edit HTML documents. However, one of the benefits of XML is pinpoint searching.

Pinpoint Searching

A major benefit of XML is the possibility for vastly improved web-search. Due to the semantic nature of XML elements, search can be restricted to elements, thereby pinpointing searches. XML searching is an acknowledged research problem and we are only beginning to explore the exact benefits of searching XML files. There are a variety of searches that can be performed in our system. (i) One can search for one or more keywords inside of one or more elements. The keywords can be treated as a conjunct by including the keyword “and” in the search term. Otherwise, the keywords are treated as a disjunct. (ii) One can search for keywords in entire documents. (iii) One can browse pages by element. In this latter case, information of the selected element is displayed for each of the documents accessible to the search engine. In all three cases, one can restrict the search to specified courses.

The type of search engine made possible by XML is advantageous for many kinds of users. Consider searching the course description pages. If a student has taken a certain course, they can search the prerequisites fields with that course’s number to see which other courses they can now take. Prospective students may search course descriptions for reference to certain programming languages.

Given information on which courses a students has taken, prospective employers can search syllabi for, say, the programming languages to which a student was exposed or the kinds of projects in which they participated. As such, XML is more than a search engine. We like to refer to these kinds of XML based search engines as *information processing engines.*

Experiences with the Technology

We performed user testing in order to obtain feedback on the usability of the editor and the search engine.

We asked about 20 students in a second computer science course to evaluate the editor. They were charged with using our editor to create XML-based syllabi, based on existing HTML-based web-pages. We asked our students to suggest improvements and comment on the editor. Their feedback indicated that for the restricted format of a syllabus, our editor works very well. Several students proposed the ability to add pictures, or to break out of the restrictions imposed by the existing XML elements. Students were not aware that we set-up our system so that one can insert any HTML tag into the editor (and hence the XML document) and that they are appropriately processed by our stylesheets. Our students furthermore suggested the ability to preview the document being edited. We since added this feature to our editor.

We asked about 30 students in a difference section of the same course to evaluate the search engine. Not all existing syllabi provide information that will soon be part of our new standard as represented in the XML documents. Hence, some of the syllabi edited as part of the prior experiment were incomplete. For this reason, we restricted this experiment to searching course description files. We asked one side of the classroom to use the XML search engine and the other one to use Google, restricted to the college's web-site. While not a perfect comparison, this set-up nevertheless gave us some interesting results.

The first observation we made is that those students who used the XML search-engine turned in the sheets with their answers before those who used Google. Some of the questions we asked were tailored to the strengths of the XML search engine. One such question was: "Which courses can I take when I have takes CS230?" Since we have an element for prerequisites, this was an easy search when using an XML search engine. Some questions, such as: "Which CS courses are required for graduation?" cannot be answered by using the XML search engine, as the course description pages do not contain any information about graduation requirements. Since we have HTML pages with that information, students using Google were able to find the answer to that question. Finally, some of the questions did not favor either search engine, such as: "What are the programming languages used in the CS curriculum?" With either search engine, students had to search for particular programming languages, and record the ones mentioned in course descriptions or syllabi.

The experiment exposed out an interesting issue. Even if information is available in XML documents, how would an XML search engine know in which documents to look for that information? The basic problem is that the XML search engine presented to our students is a special-purpose search engine, tailored to a specific class of documents. This is why it is so powerful. Based on this insight, we are now working on a general-purpose XML search engine, one in which the user enters their search string into a single text -box, but which underneath uses XML technology to provide accurate search results.

Acknowledgements

I thank Robert Signorelli for his feedback on this work as well as the students of my sections of CS220 during the fall quarter of 2002 for participating in user testing of the editor and search engine.

References

Blackboard (2002). http://www.blackboard.com/

WebCT (2002). http://www.webct.com/

Goldfarb, C.F., Prescod, P. (2000). *The XML Handbook*, 2nd Ed. Prentice Hall.

Wollowski, M. (2002). *XML Based Course Websites*. Proceedings of the 2002 E-Learn Conference.