Designing and Evaluating an Applied XML Tutorial

A Thesis
In TCC 402

Presented to

The Faculty of the
School of Engineering and Applied Science
University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

By

Kevin John Thomas

March 23, 2004

On my honor as a University student, on this assignment I have neither given nor received unauthorized aid as defined by the Honor Guidelines for papers in TCC courses.

Signed ________________________________

Approved ________________________________  Technical Advisor
Alfred Weaver

Approved ________________________________  TCC Advisor
Patricia Click
Acknowledgements

The production of knowledge involves a large number of individuals. I would like to thank my technical advisor, Professor Alfred Weaver for his guidance and technical insight throughout the course of the project. In the same way, I would like to thank my TCC advisor, Patricia Click, for helping me to uncover the social and ethical implications of this project and directing me through the thesis process as a whole. I would also like to thank all those from the Federated Trust Systems class who invested their time to test and evaluate the exercises. Last, I would like to thank my parents who made this educational opportunity possible.
Table of Contents

Acknowledgements ....................................................................................................................... i
Table of Figures .......................................................................................................................... iii
Glossary of Terms ......................................................................................................................... iv
Abstract ....................................................................................................................................... v
Chapter 1: Introduction ................................................................................................................. 1
  1.1 Purpose Statement .................................................................................................................. 1
  1.2 Background and Problem Statement ..................................................................................... 1
  1.3 Literature Review .................................................................................................................. 2
  1.4 Scope and Method ................................................................................................................ 2
  1.5 Thesis Overview ................................................................................................................... 3
Chapter 2: Review of Relevant Literature .................................................................................... 4
  2.1 Introduction .......................................................................................................................... 4
  2.2 A History of XML ................................................................................................................. 4
  2.3 The Internet as a Teaching Tool .......................................................................................... 6
  2.4 Research Gaps .................................................................................................................... 7
Chapter 3: Methods and Materials ............................................................................................... 9
  3.1 Research ............................................................................................................................. 9
    3.1.1 Initial Research .............................................................................................................. 9
    3.1.2 Resources .................................................................................................................... 9
  3.2 Activities ............................................................................................................................ 10
    3.2.1 Design ....................................................................................................................... 10
    3.2.2 Implementation .......................................................................................................... 13
    3.2.3 Testing ...................................................................................................................... 15
  3.3 Materials ........................................................................................................................... 15
  3.4 Social and Ethical Implications ........................................................................................... 16
Chapter 4: Results ......................................................................................................................... 18
  4.1 Background Information ...................................................................................................... 18
  4.2 Evaluating the Exercises and Software ............................................................................. 19
  4.3 Miscellaneous Feedback .................................................................................................... 21
    4.3.1 Suggestions for improving the content ...................................................................... 21
    4.3.2 Suggestions for improving individual exercises ....................................................... 21
Chapter 5: Conclusion .................................................................................................................. 22
  5.1 Summary ............................................................................................................................ 22
  5.2 Interpretation ....................................................................................................................... 23
    5.2.1 Achieving the project goals ...................................................................................... 23
    5.2.2 Accuracy and Completeness of the Results ............................................................ 24
  5.3 Recommendations .............................................................................................................. 25
Works Cited ............................................................................................................................... 27
Bibliography ............................................................................................................................... 28
Appendix A ................................................................................................................................... 29
Appendix B ................................................................................................................................... 34
Appendix C ................................................................................................................................... 35
Appendix D ................................................................................................................................... 39
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Screenshot of the XML Parser.</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Measuring Programming Experience.</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>Student Interest in Learning XML.</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>Quality Rating for Beginner Exercises.</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Quality Rating for Beginner Exercises.</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>Quality Rating for Intermediate Exercises.</td>
<td>39</td>
</tr>
<tr>
<td>7</td>
<td>Quality Rating for Advanced Exercises.</td>
<td>40</td>
</tr>
</tbody>
</table>
Glossary of Terms

**Standardized General Markup Language (SGML)** - SGML is an enabling technology for describing “mark up” languages. It is not a language in itself, but rather a mechanism for describing other languages. Both XML and HTML are SGML based languages.

**eXtensible Markup Language (XML)** - XML is a flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, it allows the programmer to define custom tags for describing data.

**World Wide Web Consortium (W3C)** - W3C is an institution which promotes the evolution of the Internet by developing common protocols and standards for the World Wide Web.

**Hypertext Markup Language (HTML)** - HTML is the original language for publishing information on the World Wide Web. Like XML, HTML is derived from SGML, and uses tags to mark up data.

**Document Type Definitions (DTD)** - DTD’s define the legal building blocks of an XML document. They specify which relationships are valid by specifying all the possible legal elements of an XML document.

**Schema** - A schema provides a means for defining the structure, content and semantics of XML documents. Using the Schema, a programmer can define rules and constraints on the structure of the XML document.

**eXtensible Stylesheet Language (XSL)** - XSL is a family of recommendations from the W3C for defining XML document presentation. XML describes the structure of data and XSL provides a means for displaying data.

**XSL Transformations (XSLT)** - XSLT is a standard for transforming XML into another language, for presentation or other purposes. XSLT is part of the XSL family of technologies.

**Document Object Model (DOM)** - DOM is an application programming interface that defines functions for accessing and modifying the content of XML documents. Many programming languages support the Document Object Model.

**Simple API for XML processing (SAX)** - Like the Document Object Model, SAX is an application programming interface for accessing the components of an XML document. It serially processes XML elements, but does not allow the programmer to modify the individual components of the document.

**Simple Object Access Protocol (SOAP)** - XML-based framework for exchanging structured messages between two decentralized clients across a network.
Abstract

The purpose of my thesis project work was to develop a set of laboratory exercises to teach the eXtensible Markup Language (XML). Students of CS 453, the University of Virginia’s Electronic Commerce course, have expressed interest in learning XML, and many experts consider it an essential technology for web programming. Because the current computer science curriculum lacked an educational tool to teach students about XML, this project fulfilled an actual need.

A brief historical discussion of XML reveals its impacts on the Internet and its applications in the modern programming world. The student accessed these exercises in a “virtual laboratory” environment via the Internet. Because the work involved a web based teaching tool, the thesis also explored the role of the Internet as an educational medium. This thesis project broke with other tutorials in that it attempted to present XML to the student in a broad technological perspective, focusing on applications of the language.

The process for creating the labs involved research, design, implementation, and testing. The methodology culminated with an evaluation tutorial which measured the effectiveness and quality of the project as a whole. Generally, the student testers gave the lab modules high ratings. Most labs received a score between four and five on a five point scale measuring quality.

Even with a positive rating, the project could benefit from further work. A few labs need some improvement to sustain the quality level of the whole tutorial. Students suggested some topics they would like to see covered and also gave some advice to improve individual labs. Also the software requires continuous modification to reflect the pace of technological change.
1.1 Purpose Statement

The purpose of my thesis project work was to develop a set of laboratory exercises to teach the eXtensible Markup Language (XML). The project accommodated a growing interest among computer science students at the University of Virginia to learn XML. The tutorial focused not only on teaching XML syntax, but on the practical application of XML within the domain of E-Commerce.

1.2 Background and Problem Statement

The evolution of XML as a technology reveals its potential as a powerful E-commerce tool. Originally introduced by the World Wide Web Consortium (W3C) on February 10, 1998, XML continues to transform the Internet from a publishing medium into an applications processing environment (White 4). In fact, the extensive use of XML outside the realm of the Internet establishes it as a driving force with the ability to shape future technological trends. Database programmers, web developers, and object oriented programmers each use XML as a tool for information management. XML combines extensibility and interoperability with simplicity. Its robust functionality provides the basis for countless supplementary technologies. These aspects of XML make it an essential component in any E-commerce system.

Although XML boasts an honored position as a ubiquitous web technology, the current computer science curriculum lacked an educational tool to teach students about it. In particular, Computer Science 453 (the E-Commerce course at the University of Virginia) needed new XML lab exercises to improve its content and appeal to student
interest. In addition to teaching the basics of XML, my project focused on creating an XML tutorial geared specifically towards applications in E-Commerce. The thesis work contributed to the educational content of CS 453 by creating a reusable, effective, and flexible lab module. Unlike current XML tutorials, my thesis project examined how to use XML technology to solve E-commerce problems. The ultimate goal of my research effort was to improve the student’s working knowledge of XML through exercises, examples, and software.

1.3 Literature Review

Many technical professionals and software developers recognize the impact of XML as a tool for developing E-Commerce solutions. In order to expose the historical perspective of XML, the literature review in chapter two highlights the developmental work and research that helped it evolve into such an influential piece of technology. Because the project involved a web based teaching tool, the literature review also explores the role of the Internet as an educational medium.

1.4 Scope and Method

As specified in the thesis proposal, the work for this project involved creating a single laboratory manual containing fifteen XML exercises. The lab structure, initially developed by Professor Alfred Weaver, follows specific guidelines. In particular, a complete lab divides the subject matter into three levels of difficulty. The lab structure further subdivides each level of difficulty into five exercises. All fifteen modules follow a common format, including a title that identifies the subject being introduced, an
overview of what the exercise will accomplish, example XML source code that illustrates
the point being made, a note about common errors that should be avoided, a list of
references that the student might find helpful, and finally the programming exercise itself.
Appendix A includes an example laboratory exercise. Given this prearranged structure,
my teaching tool demonstrated XML to the student in the following three levels of
increasing difficulty: the basics for creating an XML document, developing constraints
for a well formed XML document, and XML in the context of other technologies.

I accomplished the project goals throughout several steps. First, I collected the
necessary technical background to master the subject of XML, discovered applications of
XML, and looked at some of the common tools associated with this technology.
Secondly, I designed each lab, paying particular attention to the logical progression of the
material. Next, I implemented each lab along with the necessary program code and
software examples. Last, I created an evaluation survey for the students to complete to
measure the success of the tutorial. Chapter three provides greater detail concerning the
method of my thesis work.

1.5 Thesis Overview

The remainder of this thesis explains the complete scientific process of designing
and evaluating an applied XML tutorial. Chapter two discusses previous work and a
history of XML. Chapter three discusses the materials used in this thesis work and the
methodology for accomplishing the project objectives. Next, chapter four reviews the
project results. Finally, the thesis concludes with an interpretation of the results and
presents recommendations for further action.
Chapter 2: Review of Relevant Literature

2.1 Introduction

Software developers generally agree that XML is a powerful tool for developing E-Commerce applications because the technology greatly enhances the ability to exchange information. This section reviews the developmental work and research that helped XML evolve into such an influential piece of technology. Because this thesis work involved a web based teaching tool, this section also explores the role of the Internet in education. The literature review should shed light on the historical perspective of the technology, as well as its impacts and applications. In addition, it explores the educational facet of this project and examines research about using the Internet to deliver academic content.

2.2 A History of XML

XML originated from the concept of generalized markup which emerged in the 1960s. Back then, systems serving financial institutions and government required a format to model data. To meet this need, Charles Goldfarb created the Standardized General Markup Language (SGML) in 1974. As Martin Brown relates, “Everything from a full book to a quick reference card could be pulled from a raw SGML file, all without every modifying or copying the contents” (6). The robust and comprehensive nature of SGML provided a powerful mechanism for tagging data. However, developing software support for such a large language proved to be a huge programming task. As Bradley points out, “the very power of SGML has been its major handicap” (7). A need for a more simplistic approach to generalized markup arose from the deficiencies of SGML.
Tim Berners-Lee, creator of the World Wide Web, adopted a solution to this problem. In 1989, he used the basic mechanisms provided by SGML to create a subset of the language call Hypertext Markup Language (HTML). HTML provided a way to mark up text for displaying it through a web browser. As the web began to grow in popularity during the 1990s, HTML was extended to meet the needs of a growing audience. However, as Daconta remarks, “proprietary extensions to HTML were counter-productive and ill-suited to general use” (8). Also, scalability issues arose from the fact that users could not extend HTML by adding new tags. The Internet now required a simple, extensible, open standard for modeling information.

Technologist began to explore the relationship between SGML and HTML for answers. As Brown notes, “It became apparent that the same principles that applied to SGML and HTML could be applied to modeling data” (6). In 1997, the World Wide Web Consortium established 10 design objectives that a new language should fulfill. Gulbransen writes that, “The goal of the original working group was “80/20,” that is, to get 80% of the functionality of SGML with only 20% of the complexity.” The goals ensured that the new language should be simple, compatible, practical, and easy to use. By February 10, 1998, W3C had adopted, implemented, and released a language that satisfied these requirements: XML version 1.0. Although many of the features remain unrealized, XML has evolved into the preferred standard for storing data.

Since its inception, XML has grown into a wide family of technologies. Schema and Document Type Definitions ensure that XML syntax adheres to particular rules. eXtensible Stylesheet Language (XSL) and XSL Transformations (XSLT) help to format XML for displaying the data as a web page. Many programming languages support the
Document Object Model (DOM) and Simple API for XML processing (SAX). These technologies provide software support for reading, manipulating, and creating XML files. Finally, several emerging remote technologies, including Remote Procedure Call (RPC) and Simple Object Access Protocol (SOAP), use an XML core to enhance communication between computers. Indeed, this language has great potential to transform our information age.

2.3 The Internet as a Teaching Tool

Educational research has reveals the potential of the Internet as a medium to convey academic material. Recent findings dispel several myths that negatively affect the adoption of electronic learning. However, research also exposes several cautions in using an online academic environment.

Some common beliefs about how the Internet affects education discourage support for delivering academic content via the World Wide Web. First, some assume that the Internet used as a means of education may lead to the death of the campus university. On the contrary, as Weller points out, “there is more to a university education than just a degree” (27). Colleges offer a social atmosphere, inherent discipline for studying, and most importantly, human interaction with real professors. In the same way, academics argue that the Internet will lead to declining standards in education. In fact, the opposite holds true. Tom Russels’ book, The No Significant Difference Phenomenon, concludes that the content of educational material, not the mode of delivery, most affects student performance. Debunking these myths demonstrates that the Internet can be an effective educational technology.
However, with the decision to use the Internet as a teaching tool, one should still exhibit caution. In particular, using the Internet as a “mass medium” (Weller 16) or academic distribution mechanism to increase the instructor to student ratio makes for a poor educational experience. Similarly, personal interaction must still play an important role in the academic process, lest, as Partee suggests, a student becomes lost behind a “dazzling shower of technologically generated images and data” (56). Moreover, the online content should be easily modified and open to change. These areas force the educator to use vigilance when implementing an online learning environment.

Despite these cautions, the Internet lends great potential to the discipline of education. As Partee notes, the Internet meets the “increasingly extensive demands for knowledge in a sophisticated society” (18). Online learning is not just a passive stream of information because it engages the student and forces interactivity with the academic content. Although the scope of my project does not focus on determining the effectiveness of an online educational environment, these issues are still pertinent given the web based nature of the XML tutorial. With respect to this project, I believe that the pedagogical framework of the virtual E-commerce laboratory satisfies the requirements to successfully use the Internet as an effective teaching tool.

2.4 Research Gaps

Most research concerning XML has focused on expanding the functionality or reducing the complexity of the language. The work done thus far concentrates on the potential of the technology and not the application of XML within a specific context.
This educational project extended from the work of previous XML researchers and used their foundations to teach students the basics of XML, but more importantly, its applications and ethical implications.

Although many computer science educators have already created a plethora of XML tutorials and training documents, they usually focus on specific features of the language or on a specific technology. As a result, these tutorials are not specially geared toward applications development in the context of E-Commerce. This thesis project aimed to present XML to the student in a broad technological perspective. Thus, the student developed a variety of methods to approach E-Commerce problems and mastered multiple tools for creating powerful E-Commerce solutions rather than simply learning the language.
Chapter 3: Methods and Materials

3.1 Research

3.11 Initial Research

The first step in completing my thesis project involved collecting the necessary technical background to master the subject of XML. In this phase of research, I learned XML programming, the applications of XML, and how to use some of the common tools associated with this technology. After solidifying the thesis project idea with my technical advisor, I studied XML from the last week of August 2003 until the end of September 2003. The initial research revealed a broad range of XML applications and a jungle of technologies to supplement XML’s functionality. Thus, my research also involved narrowing down the scope of the project and picking certain aspects of XML on which to focus.

3.12 Resources

I used several different types of sources to aid my research effort. The Science and Engineering Library provided an extensive collection of technical literature on XML. Because the printed word often lags behind rapid technological changes, I used the Internet to access up to date information on technological specifications. Table one describes the main websites I used in researching my thesis.

<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/">http://www.w3.org/</a></td>
<td>The World Wide Web Consortium homepage. It has the most up to date specifications for the technologies they release.</td>
</tr>
<tr>
<td><a href="http://www.w3schools.com">http://www.w3schools.com</a></td>
<td>Contains several useful XML tutorials.</td>
</tr>
</tbody>
</table>
Table 1: This table provides a list of references used in the research phase of the thesis work.

<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.saxproject.org/">http://www.saxproject.org/</a></td>
<td>Contains information related to XML parsing, specifically the Simple API for XML parsing.</td>
</tr>
<tr>
<td><a href="http://msdn.microsoft.com/">http://msdn.microsoft.com/</a></td>
<td>The Microsoft developer’s network provides useful information on how to program and use different languages.</td>
</tr>
<tr>
<td><a href="http://iis.cs.virginia.edu/webweavers/ee%20labs/">http://iis.cs.virginia.edu/webweavers/ee%20labs/</a></td>
<td>This is part of the class webpage for Electronic Commerce, CS 453. It contains all of the lab exercises.</td>
</tr>
</tbody>
</table>

3.2 Activities

3.21 Design

The labs for CS 453 already had a particular structure set forth by the Professor, Alfred Weaver. See the “Scope and Method” section of chapter one for a description of the lab structure. With the methods of presenting information to the student already situated, the largest part of the design phase involved choosing specific content for the individual sections. My preliminary XML research facilitated the process of deciding the technologies most beneficial to a student studying E-Commerce. In addition to choosing content, I had to classify the material according to level of difficulty. For this task of grouping by difficulty, I looked at the previous labs from the class website. These labs provided insight on what type of material constituted a certain difficulty level. In the same way, I wanted to group exercises together that illustrate a certain technical theme. With these requirements, I set out to mold my research into a useful XML tutorial

I devised a scheme that classified topics both by level of difficulty and adherence to a certain theme. The beginning section, entitled “the Basics of Creating an XML Document,” as its name implies, introduced the student to basic XML concepts. I chose
to focus on certain aspects of the XML language. Specifically, I decided to teach the student the most fundamental components of XML including elements, entities, attributes, and the CDATA section. After providing the student with the necessary background, I discussed basic rules concerning data modeling decisions. Last, to bridge the conceptual gap between HTML and XML, I presented a way for the student to render the data represented by an XML document.

The intermediate section, titled “Developing Constraints for a Well Formed XML Document,” taught the student how to form restrictions on the structural representation of XML data. A Document Type Definition (DTD) defines the structure of an XML document by specifying legal elements, element contents, and a list of attributes and restrictions for elements. The first two exercises taught the student how write basic constraints using a Document Type Definition. The third exercise demonstrated how to link a Document Type Definition to an XML file. The last two exercises discussed a slightly more advanced method of constraining an XML document known as XML Schemas. Schemas offer advanced functionality and more flexibility than DTD’s.

The advanced section, titled “XML in the Context of Other Technologies,” revealed how XML applications work within the framework of other tools. In fact, the advanced section distinguishes this tutorial from other XML lessons. It demonstrates specific applications of XML and shows how XML serves as a functional basis for other technologies. I decided to introduce the students to the simple API for XML processing (SAX), the document object model (DOM), extensible style sheet language (XSL), the simple object access protocol (SOAP), and XML web services in visual studio .NET. Refer to the glossary of terms for a specific definition which details the functionality of
these technologies. Table two lists every exercise contained in the XML tutorial and briefly explains the education goals of the exercise.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Number</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginner</td>
<td>1</td>
<td>Basics of Creating an XML document</td>
<td>Basic components of an XML document</td>
</tr>
<tr>
<td>Beginner</td>
<td>2</td>
<td>Using XML Parsers</td>
<td>Creating and parsing XML documents</td>
</tr>
<tr>
<td>Beginner</td>
<td>3</td>
<td>Advanced XML Components</td>
<td>More components of XML</td>
</tr>
<tr>
<td>Beginner</td>
<td>4</td>
<td>Attributes vs. Elements</td>
<td>Tips for modeling XML data</td>
</tr>
<tr>
<td>Beginner</td>
<td>5</td>
<td>Displaying XML</td>
<td>Basic methods for displaying XML data</td>
</tr>
<tr>
<td>Intermediate</td>
<td>6</td>
<td>Legal Building Blocks of XML</td>
<td>Building constraints for XML documents with DTDs</td>
</tr>
<tr>
<td>Intermediate</td>
<td>7</td>
<td>Advanced DTD’s</td>
<td>Advanced functionality of DTDs</td>
</tr>
<tr>
<td>Intermediate</td>
<td>8</td>
<td>Linking DTD and XML</td>
<td>Combining a DTD with an XML document</td>
</tr>
<tr>
<td>Intermediate</td>
<td>9</td>
<td>W3 XML Schemas</td>
<td>Defining the semantics of XML documents</td>
</tr>
<tr>
<td>Intermediate</td>
<td>10</td>
<td>Advanced Schemas</td>
<td>Advanced features of Schemas</td>
</tr>
<tr>
<td>Advanced</td>
<td>11</td>
<td>Extensible Stylesheet Language</td>
<td>A more robust method for displaying XML data</td>
</tr>
<tr>
<td>Advanced</td>
<td>12</td>
<td>Simple API for XML Parsing</td>
<td>A method for using XML data with processing applications</td>
</tr>
<tr>
<td>Advanced</td>
<td>13</td>
<td>The W3C Document Object Model</td>
<td>An alternate programming interface for processing XML data</td>
</tr>
<tr>
<td>Advanced</td>
<td>14</td>
<td>Simple Object Access Protocol</td>
<td>A real world example of a messaging framework which uses XML</td>
</tr>
</tbody>
</table>
3.22 Implementation

The implementation phase followed the design. Implementation involved writing the actual exercises and accompanying software to demonstrate the concepts within the exercises. The tutorial often required that the student take the provided software and modify it as an exercise. Thus, the example software provided a basis from which the student completed the exercises. With the example software, the student spent more time learning XML and less time implementing software from scratch.

Because the tutorial contained many example XML files, I developed an extremely important tool for the student called an XML parser. A parser provides a mechanism for the programmer to verify that the XML code is syntactically correct and it also checks XML code against the rules defined in a Document Type Definition. It was a valuable tool for students beginning XML programming but also was useful for the more advanced exercises. With accessibility and user friendliness in mind, I programmed the XML parser as a web page using Java Script. Figure one shows a screenshot of the XML parser in action.
The advanced section, because of its difficulty and application oriented approach, required the most software. For the advanced section I coded a web page in JavaScript that used the DOM to extract information from an XML file. For the exercise on the Simple API for XML parsing, I used the Java programming language to create a test program that demonstrated the main features of SAX parsing. I also wrote a simple web service in C# (pronounced C-Sharp) using Visual Studio .Net to expose the role that XML plays in web services. These pieces of software provided key examples and helped to further illustrate the concepts presented in the exercises.
3.23 Testing

After creating the labs, I needed to measure their ability to successfully teach the student about XML. I had a meeting with my technical advisor and arranged a time slot during his class (CS 551: Federated Trust Systems) to give a short presentation on my thesis work and walk through some example lab exercises. After the short pitch that I made to the class, I handed out a sign up sheet for students to volunteer their time in evaluating the exercises. The students signed up to complete five exercises, or a single section difficulty of the tutorial. I made an electronic survey using the Survey Suite software that Professor Weaver created as a research project. The students who signed up to test the labs then completed the survey. The survey can be found on the Internet: http://intercom.virginia.edu/SurveySuite/Surveys/XMLEvaluation/. A text version of the survey is also included in Appendix C.

3.3 Materials

The project required several pieces of software and a computer to successfully complete the objectives. I used the Internet to research and find web based tutorials for the technologies I chose to present. I also made extensive use of the Visual Studio .NET development platform to author the XML files. In addition to using Visual Studio .NET, I used a separate development environment called JCreator LE to write the Java source code for the advanced section. Both of these development environments provided a useful syntax highlighting feature. For some types of files, such as the Cascading Style Sheets and Schemas, I used simple text editing software. Outside of the development
environments, I tested the software using Internet Explorer. With these simple tools, I created the fifteen lab exercises.

3.4 Social and Ethical Implications

I created a foreword to the tutorial to highlight some of the social, ethical and organizational implications of using the eXtensible Markup Language. When dealing with a particular social or ethical issue, the foreword encourages the reader to explore other XML tools related to the problem. It also discusses the organizational impacts of XML as background information for the reader. This tutorial fosters an ethically informed learning environment by providing the information before the student starts learning XML.

The ethical implications of this project lie not within XML itself but within the scenarios and abilities that this standard enables. This technology opens doors to the visually impaired or illiterate World Wide Web surfer. VoiceXML allows the computer to read a page to the user and even incorporates voice recognition to facilitate web surfing. Because this tutorial does not cover VoiceXML, the foreword points the reader to the W3C Voice XML tutorial to discover the full capabilities of the technology.

Likewise, companies and organizations can now exchange data quickly and in massive quantities. Issues arise with user privacy, and data theft. A stream of bits hides the meaning of information stored in a database; however, the eXtensible Markup Language tags data in plain text. As a result, one must pay particular attention when exchanging unencrypted XML documents. The W3C has addressed this problem by releasing standards for XML encryption. The foreword directs the reader to the XML
Encryption Working Group who has released several specifications for encrypting XML syntax.
Chapter 4: Results

As stated in the methodology section, I created an online survey to evaluate my XML tutorial. The background section of the survey reinforced one of the initial motivations for undertaking the project, to fulfill an actual need. I also designed the survey so that the responses would reflect both how effectively the labs conveyed the material and the quality of the accompanying software and examples. Moreover, the survey responses included general feedback on the topics covered and suggestions for improving individual exercises.

4.1 Background Information

The questionnaire initially revealed information about the student’s personal interests and programming background. The first question attempted to determine the student’s XML programming proficiency. The results from this question, presented in figure two, showed that students generally had little experience with developing XML documents.

![Student's Previous Experience With XML]

Figure 2: This chart shows the level of XML experience students had before completing the exercises.
Also, students responded positively to the question asking about their interest in learning XML. The outcome of this question, illustrated in Figure three, revealed that a majority of the students were interested in learning XML while the other expressed either an extreme or moderate interest in learning XML. Similarly, the students surveyed unanimously agreed that the value of knowing XML is enough to warrant studying the technology with respect to other E-Commerce topics. These initial background questions emphasized the need for undertaking this project.

![Student Interest in Learning XML](image_url)

Figure 3: This chart shows student interest in learning XML.

### 4.2 Evaluating the Exercises and Software

The next part of the evaluation tool solicited data concerning the quality of the lab exercises, software, and examples. The first question asked the student to rate on a scale of one to five how helpful the lab exercises were in learning the subject material. Five indicated that the lab effectively taught the lab material and one indicated that the lab poorly presented the ideas and topics. Similarly, the questionnaire asked the testers to
rate the quality of the software and examples for each exercise reviewed. Once again the students used a five point scale, five indicated “excellent” quality and one indicated “no value whatsoever.” In order to legitimize the results, the survey also asked the tester the degree to which he or she reviewed the exercises. The results show that the students thoroughly reviewed the material and thus were in a strong position to critique the labs. With few exceptions, the students gave the exercises high marks. At the negative end of the spectrum, the students did not categorize any of the software as “minimal value” or “no value whatsoever.” Figure 4.3 illustrates the average exercise and software ratings for the five beginning level exercises.

![Average Exercise and Software Rating](image)

**Figure 4:** Average exercise and software quality ratings for the beginning level exercises.

On the whole, the testers considered the software and examples a valuable addition to the tutorial. The ratings for the intermediate and advanced exercises are included as a part of Appendix D.
4.3 Miscellaneous Feedback

4.31 Suggestions for improving the content

At the end of the survey the student had the opportunity to comment on the topics covered by the individual exercises. The responses revealed a similar trend in that the students wanted to focus on creating XML constraints using Schemas rather than Document Type Definitions. Schemas address many of the shortcomings of Document Type Definitions and offer advanced functionality. In the same way, one student suggested that the labs include a section on XPath technology. XPath is a major element of the eXtensible Stylesheet Language that helps to define parts of an XML document. These suggestions perhaps were motivated by a desire to focus on more practical technologies. These modifications could easily be integrated into the tutorial given its flexible design.

4.32 Suggestions for improving individual exercises

The testers also made suggestions for improving the quality of individual exercises. One student proposed that some of the exercises, especially those dealing with more complicated subject material, could use more examples. Another student suggested not including the full specifications for the technologies being taught. These specifications are often extremely detailed, long, and therefore may not be the most helpful references for the student. One student commented that the “Displaying XML” exercise was complicated and that the example did not help illustrate how to complete the exercises. These suggestions highlight a few minor flaws in an overall successful tutorial.
Chapter 5: Conclusion

The project objectives included teaching XML, creating an effective reusable lab module, and enhancing the educational content of the computer science curriculum. The results provide a measurement of how well these objectives were accomplished. The conclusion first summarizes the results. With a factual summary, the interpretation section next discusses the significance of certain results. The last section presents recommendations for further actions based on the interpretation of the results.

5.1 Summary

The questionnaire asked a few questions concerning the student’s programming background and interests. In general students had little experience programming with XML and therefore lacked proficient XML skills coming into the exercises. Despite their lack of experience, students expressed a keen interest in learning XML. Similarly, the students surveyed unanimously agreed that the value of knowing XML is enough to warrant studying the technology with respect to other E-Commerce topics. These responses work to establish reasons for completing the project, particularly, to teach the students an important skill that they know little about but are interested in learning.

In addition to revealing information about programming proficiency and interest, the responses reflect the quality of the tutorial as a whole. The survey shows that students comprehensively completed the exercises, with few exceptions. A majority of the exercises received a rating of four or five on a scale of one to five (five designates excellence). The scores for the software follow the same positive trend as the scores for the exercises. Generally, the exercises and software were well received by the students,
thoroughly reviewed, and scored high marks in quality and effectiveness in conveying the material.

5.2 Interpretation

5.2.1 Achieving the project goals

One of the main objectives of the thesis work was to improve the educational content of CS 453 by fulfilling a real need while catering to student interests. The survey demonstrated that students lack experience in developing XML documents while the introduction established XML as a critical E-Commerce technology. One can conclude therefore that the computer science curriculum was missing a vital educational component. In fact, the students collectively agreed on the importance of studying XML in comparison to other technologies. Moreover, students expressed an interest in learning XML. Thus, this project worked not only to meet a real need but to satisfy student interest. The results of the survey show that the project achieved its goal in uniting a vital E-Commerce programming skill with a unanimous interest.

The ultimate goal of this project was to improve the student’s working knowledge of XML through exercises, examples, and software. The students responded positively to most of the exercises, ranking them with a four or five. The scores reflected not only the quality of the labs but effectiveness in conveying the material and relevance to the topic. Indications that the students performed a thorough review of the exercises further legitimize the results. Once again the results show that the project has realized its objective to deliver quality exercises effective at conveying material.
Three of the exercises, however, received a slightly lower mark than the others. The exercises “Displaying XML,” “Linking DTD and XML,” and “Simple Object Access Protocol” received a score of three at least once during the rating process. Several reasons may have contributed to the lower score. It is possible that the students thought the content irrelevant. Perhaps the students considered the content relevant but ineffective in teaching the subject. Likewise, the students may not have perceived the difficulty of the exercises as intellectually stimulating. Irregardless, these individual exercises require attention in order to maintain the quality of the tutorial as a whole.

5.22 Accuracy and Completeness of the Results

For several reasons, the evaluation process lent itself to anecdotal evidence rather than scientific method. First, only a small group of people tested the material. Although the Federated Trust Systems class at UVA provided testers with the necessary skills, it was difficult to entice them to complete the survey without any incentive. The class has about eighteen students and 33% responded to the survey. Second, the evaluation process focused on the quality of the material and did not quantitatively measure the XML knowledge of the student through a testing procedure. Although a small testing pool and a quality-focused survey still provided solid feedback about the success of the project, the process lacked a scientific basis.

Similarly, the structure of the survey hindered parts of the improvement process. One fault of the survey process was that it did not reveal the exact faults for individual exercises. The survey provided a general feedback section for comments relating to specific exercises. When students rated an exercise low, they did not fill in the section to explain why the material scored poorly. The general comments provided no clues as to
the reason these exercises did not perform as well as the others. Thus, the evaluation lacked a mechanism to determine what the exact faults are with individual exercises.

5.3 Recommendations

Most importantly the evaluation process needs a scientific process for evaluating the labs. In addition to a survey, the final evaluation may benefit from two XML tests. The student could complete a test before reviewing the tutorial. Then as a quantitative measure, a final test could score the students XML knowledge. Comparing these two tests for each student provides numerical evidence for the effectiveness, relevancy, and quality of the lab exercises. The tests could also be integrated into a class to count as a small portion of the grade; thus providing incentive for the student to evaluate. Both a scientific process and an incentive for a large number of people to complete the tutorial could readily improve the evaluation process.

A few of the lab exercises need modification to meet a score of excellence. Changing the lab modules requires a method for discovering the deficiencies of these exercises and concentrating on fixing them. One could examine the feedback for improving certain labs then implement the changes. Alternatively, student feedback regarding topic coverage may provide a solution. One could investigate student ideas for topics and perhaps incorporate them into the tutorial, replacing one of the low scoring exercises entirely. In particular, students express interest in Schema, and want to focus less on DTDs. One student expresses an interest to see XPath technology discussed as an exercise. Fixing the negative aspects of the exercises or replacing them entirely could improve the content of the tutorial.
Another area of additional work includes the software and examples. Internet
based technology changes rapidly. The examples and software contained within the
tutorial reflect current trends in web programming. Programming language features are
constantly updated as new technologies and methods emerge for solving problems.
Accordingly, the software requires continuous refinement to keep it in tune with
technological changes.
Works Cited


Bibliography


Appendix A

Intermediate XML

Exercise 5 – Advanced Schemas

Overview

The advanced functionally of schemas provides additional incentive for choosing them over document type definitions. Specifically, schemas offer explicit control for restricting the values that elements and attributes can assume. Also, schemas provide more options for specifying the grouping and cardinality of elements. Schemas even support a notion similar to inheritance by allowing the programmer to create a complex type as an extension of an existing complex type. As programmers use XML for increasingly complex applications, schemas fulfill the shortcomings of DTD’s in validating data.

Sample Code: Building a Schema

Example 1: Restrictions on Values

- The programmer can restrict the value of elements or attributes by using inclusive values, enumeration, and regular expressions. Although constructing regular expressions resides outside of the scope of this tutorial, they offer a powerful method for forming pattern constraints for strings. Each of these methods uses the restriction and simpleType elements.

```xml
<!--A restriction using inclusive values and number of digits-->
<!--socialSecurity must have exactly nine digits-->
<xsd:element name="socialSecurity">
  <xsd:simpleType>
    <xsd:restriction base="xsd:integer">
      <xsd:minExclusive value="000000000"/>
      <xsd:maxInclusive value="999999999"/>
      <xsd:totalDigits value="9"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- A restriction example using enumeration -->
<xsd:element name="sport">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Hockey"/>
      <xsd:enumeration value="Golf"/>
      <xsd:enumeration value="Football"/>
      <xsd:enumeration value="Basketball"/>
      <xsd:enumeration value="Baseball"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```
Example 2: Ordering elements with grouping

- Schemas have grouping constructs called indicators which specify how XML elements are ordered in a document. The three grouping elements in schema include all, choice, and sequence. The xsd:all group indicates that each XML element contained within its span occurs at most one time but that the elements may occur in any order.

```
<xsd:element type="openPositions"/>
<!-- Elements may occur in any order -->
<xsd:complexType name="openPositions">
  <xsd:all>
    <xsd:element name="programmer" type="xsd:string"/>
    <xsd:element name="tester" type="xsd:string"/>
    <xsd:element name="analyst" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

- The xsd:choice group indicates that either one element or another can occur

```
<xsd:element type="address"/>
<!-- The address tag includes only one of these elements -->
<xsd:complexType name="address">
  <xsd:choice>
    <xsd:element name="home" type="xsd:string"/>
    <xsd:element name="work" type="xsd:string"/>
    <xsd:element name="school" type="xsd:string"/>
  </xsd:choice>
</xsd:complexType>
```

- The xsd:sequence group indicates that each element appears exactly once in the specific order listed within the span. Using minOccurs and maxOccurs as an attribute for each element numerically specifies the cardinality of each element. This option yields more precise control than DTD’s with respect to the number of times an element can occur.

```
<xsd:element name="personName" type="nameType" />```
Example 3: Deriving Types by Extension

- When a complex type is derived by extension, its content model consists of the content model of the base type plus the content model declared in the type extension. In other words, a derived complex type inherits all the elements and attributes of the base complex type and also has newly defined elements and attributes declared using the extension and complexContent elements.

Cautions
1. For the all, choice, and sequence indicators the default values for the maxOccurs and minOccurs attributes are both equal to “1.”
2. When using the xsd:all grouping element, the minOccurs attribute can be set to 0 or 1, but due to the nature of the construct, the maxOccurs can only be set to 1.
3. xsd:all can contain only individual element declarations, not xsd:choice or xsd:sequence elements.
4. Attribute declarations must reside outside the span of grouping constructs. Attribute occurrence is governed by the use attribute.
5. There are many empty elements in W3C XML schema language: remember to close all tags explicitly or to end them with a backslash.

Reference

- Same references as previous section

Exercises

1. Use the restriction element to constrain the values of the following simple elements.
   - A book chapter that has between 2 and 30 pages.
   - A social security number that must have the form xxx-xx-xxxx. Each character must be a digit. (hint: build a regular expression)
   - A video game element that must have one of three possible video game titles: Battlefield 1942, Warcraft 3, and Grand Theft Auto.
   - A person’s initials which must be only 3 capital alphabetic letters, no numbers.
   - A number that must have 6 digits.

2. Derive a complex type from the base type in the following schema code. The type should be a more specific car, perhaps with attributes and information on the make and model.

```xml
<xsd:element name="car" type="cartype"/>
<!--Base type -->
<xsd:complexType name="cartype">
    <xsd:sequence>
        <xsd:element name="type" type="xsd:string"/>
        <xsd:element name="color" type="xsd:string"/>
        <xsd:element name="year" type="xsd:qYear"/>
        <xsd:element name="transmission" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
```
3. Write a complex element schema declaration that constrains this XML code. Use grouping to achieve the desired ordering and cardinality.

```xml
<Product productId="12987">
  <!--Elements must occur in exactly this order-->
  <Item>Shoe</Item>
  <!--Programmer must choose either designer or manufacturer-->
  <designer></designer>
  <manufacturer></manufacturer>
  <!--A shoe can have up to three colors-->
  <Color>Red</Color>
  <Price>129.99</Price>
  <Style>Cross Training</Style>
  <releaseDate>2003-11-17</releaseDate>
</Product>
```

4. Explain how the choice grouping element differs from restrictions provided by the enumeration element.
Appendix B

Source Code for the XML Parser

<!-- Kevin J Thomas September 23, 2003-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Microsoft XMLDOM parser</title>
<script language="JavaScript">
function validate() {
    var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
    xmlDoc.async="false"
    xmlDoc.loadXML(document.validator.xmltext.value)

    if(xmlDoc.parseError.errorCode!=0) {
        //document.write(xmlDoc.parseError.errorCode)
        document.validator.code.value = xmlDoc.parseError.errorCode;
        document.validator.reason.value = xmlDoc.parseError.reason;
        document.validator.line.value = xmlDoc.parseError.line;
    } else {
        window.alert("No Errors!");
        document.validator.code.value = "";
        document.validator.reason.value = "";
        document.validator.line.value = "";
    }
}
</script>
</head>
<body>
<center>
<h1>Validate XML with a Parser</h1>
<form name="validator" action="">
<textarea name="xmltext" rows="15" cols="50">
</textarea>
<br><br>
<input type="button" value="Validate" onclick="validate()">
<p>Error Code: <input name="code" type="text" size="40"></p>
<p>Error Line Number: <input name="line" type="text" size="4"></p>
<p>Error Reason: <input name="reason" type="text" size="50"></p>
</form>
</center>
</body>
</html>
Appendix C

XML Lab Exercises Evaluation Survey

1. Using a scale of 1-5, please state how proficient you were with XML before you looked at the lab exercises.

   1—no knowledge
   2—knew a little but had not written XML documents before
   3—had done some developing with XML
   4—experienced XML developer
   5—expert XML developer

2. With respect to Electronic Commerce and Federated Trust Systems, state your level of interest in learning about XML and related technologies.

   1 -- no interest whatsoever
   2 -- minimal interest
   3 -- interested in learning XML
   4 -- moderate interest
   5 -- Extreme interest

3. Do you think the value of knowing XML is enough to warrant studying it in comparison with other electronic commerce technologies (yes/no)?

   Yes   No   Don’t know

4. Please evaluate the labs you reviewed. For each lab, first state how much of the lab you completed by circling one of these options.

   A—did not review it
   B—read it, but did no programming
   C—read it and programmed at least one exercise
   D—read it and programmed all exercises.

   Then, for those labs that you did review, please rate on a scale of 1-5 how helpful that lab was in learning the subject material.

   1—no value whatsoever
   2—minimal value; it exposed me to the concept, but I still don’t understand it
   3—worthwhile; I understand the concept and can program it
   4—valuable; I am now fluent in that concept
   5—excellent; now I am the master of that concept
Then indicate whether you looked at the “answer” portion of that particular lab.

<table>
<thead>
<tr>
<th>Lab</th>
<th>Your level of completion—circle one</th>
<th>Your rating of this lab—circle one</th>
<th>Did you use “answers”?</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1 - Basics of Creating an XML document</td>
<td>A B C D</td>
<td>1 2 3 4 5</td>
<td>Yes No</td>
</tr>
<tr>
<td>B2 - Using XML Parsers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3 - Advanced XML Components</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4 - Attributes vs. Elements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5 - Displaying XML</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1 - Legal Building Blocks of XML</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2 - Advanced DTD’s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3 - Linking DTD and XML</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I4 - W3 XML Schemas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5 - Advanced Schemas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1 - Extensible Stylesheet Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2 - Simple API for XML Parsing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3 - The W3C Document Object Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4 - Simple Object Access Protocol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5 - An Introduction to XML Web Services</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Please evaluate the quality of the software and examples contained within the “extras” folder. For each lab, please rate on a scale of 1-5 the value of the software and examples in learning the subject material.

1—no value whatsoever; software did not help to illustrate concept
2—minimal value; it exposed me to the concept, but I still don’t understand it
3—worthwhile; software helped to establish the concept
4—valuable; software helped to establish the concept reasonably well
5—excellent; software helped establish concept exceptionally well

<table>
<thead>
<tr>
<th>Lab</th>
<th>Quality of software—circle one</th>
<th>General comments relating to software and examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1  - Basics of Creating an XML document</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>B2  - Using XML Parsers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3  - Advanced XML Components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4  - Attributes vs. Elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5  - Displaying XML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1  - Legal Building Blocks of XML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2  - Advanced DTD’s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3  - Linking DTD and XML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I4  - W3 XML Schemas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5  - Advanced Schemas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1  - Extensible Stylesheet Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2  - Simple API for XML Parsing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3  - The W3C Document Object Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4  - Simple Object Access Protocol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5  - An Introduction to XML Web Services</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. With reference to any particular lab, how would you suggest that it be improved? Please comment on whether any part of the lab was unclear, or what portions might need elaboration or a better explanation.

<table>
<thead>
<tr>
<th>Lab number (e.g., B1, I2, A3)</th>
<th>Suggestions for improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td></td>
</tr>
</tbody>
</table>
7. Are there XML topics or related technologies not covered in the lab that you think should be covered, either within an existing exercise or in an entirely new exercise? If so, what topics?

8. What else would you like the author of the lab exercises to know?
Appendix D

Results from the Evaluation Survey

**Average Exercise and Software Rating**

![Chart](chart.png)

**Figure 5:** Average exercise and software rating for beginning exercises.

---

**Average Exercise and Software Rating**

![Chart](chart2.png)

**Figure 6:** Average exercise and software rating for intermediate exercises.
Average Exercise and Software Rating

Figure 7: Average exercise and software rating for advanced exercises.