Summer 2005

Services Learning Practicum

Andrew Charles Swanson
Regis University

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Acknowledgements

Thanks to Hal Friskey for reviewing this paper. Thanks to Trisha Litz for her contribution to this whole project. She provided the business requirements and organized the project as faculty advisor. Trisha Litz and Wolf Thompson provided good advice and feedback on the original design and architecture. Julie Thomas contributed early on in the project, providing feedback on the database design.

Thanks to the Director of Testing and Quality Assurance, Teri Lane and my fellow employees of GuideWorks, LLC, who supported me and provided me access to Professional Office 2003 and Microsoft .Net software development tools as part of the ongoing efforts at GuideWorks to help their employees pursue their academic goals.

Most of all, I would like to thank my family for the support they have shown over the past few years while I have been studying. They have tolerated my attention to academic and professional goals when I was also needed as husband, father, and grandfather. My wife, Kathleen, has been the greatest support to me and I could not have done any of this without her.

Abstract

The Service Learning Practicum is a document driven knowledge management system. The purpose of the Practicum is to involve MSCIT students in IT projects that support and provide solutions to non-profit organizations or non-governmental organizations (NGOs). The main documents that drive the practicum are the student application; the NGO needs statement; and the student's thesis, design, and research documents. The practicum unites graduate students with NGOs. Both benefit from this union because the student is able to reach academic goals and the NGOs are able to implement low-cost or no-cost solutions for their IT needs. This project implements a graphic user interface for the collection, storage, and access of these documents.

Table of Contents

Revision History 4
Acknowledgements 5
Abstract 6
Table of Contents 7
List of Tables 9
List of Figures / Exhibits / Addenda 10
Chapter One: Introduction / Executive Summary 11
  Statement of the problem and goal to be achieved 11
  Needs 11
  Barriers 12
  XML solutions to be discussed 12
  Project Scope 14
  Definition of terms 14
  Summary 16
Chapter Two: Review of Literature / Research 17
  Overview of literature and research on XML 17
  Project research 18
  Summary: what is known and unknown about XML 21
  Contributing aspects of this project 23
Chapter Three: Methodology 25
  Research methods to be used 25
  Life-cycle models to be followed 26
  Specific procedures 28
  Formats for presenting results/deliverables 30
  Review of the deliverables 30
  Resource requirements 30
  Outcomes 31
  Summary 32
Chapter Four: Project History 35
  Project Origin 35
  Project Management 35
  Project Milestones 36
  Changes to the project plan 36
  Evaluation of project goals 37
  Discussion: Project achievement and frustrations 38
  Discussion: Impact of project variables 38
  Findings/analysis 39
Chapter Five: Lessons Learned / Next Evolution 40
  The project experience 40
  What could have been done differently 40
  Discussion: achievement of initial project expectations 41
  Next stage of project evolution 42
  Conclusions/recommendations 43
  Summary 44
Works Cited 47

Addendum A: Business Implementation..... 49
Addendum B: XML Schema..... 56
Addendum C: Project Schedule..... 64

List of Tables

Table 1 Revision History..... 4
Table 2 XML Product Solutions..... 13
Table 3 XML Engine selection..... 18
Table 4 Project Schedule..... 64

List of Figures / Exhibits / Addenda

Figure 1 Rapid Prototype Diagram (Schach 71)..... 27
Figure 2 Amaya..... 29
Figure 3 InfoPath..... 32
Figure 4 Business Use Case Diagram..... 49
Figure 5 MVC Architecture..... 54
Figure 6 Workbook, workflow ERD..... 56
Figure 7 SLP Schema..... 62
Figure 8 Schema converted with MS .NET..... 63

Addendum A: Business Implementation..... 49
Addendum B: XML Schema..... 56
Addendum C: Project Schedule..... 64

Chapter One: Introduction / Executive Summary

Statement of the problem and goal to be achieved

Research thesis:

A knowledge management system can be designed and implemented, using little or no cost Extensible Markup Language (XML) technology, to unite Regis University graduate students seeking research projects with non-profit organizations seeking engineers to implement computer and information technology projects.

Needs

Each year, non-profit organizations or non-governmental organizations (NGO) are faced with technology hurdles. Some wish to create databases to retain valuable information about their services and their organization. Some wish to create Web sites on the Internet in order to provide information about their services. Still others wish to get newly acquired hardware or software up and running. And finally some leaders in these organizations are so busy with day to day work that they cannot find the resources to even consider how technology can help them. On the other hand, Regis University students in Computer Science need real and practical problems to study. Most academic problems in computer science involve fictional

companies or organizations with fictional problems. The Services Learning Practicum suggests a means to provide problems posed by NGO needs and students who can work to find computer science solutions. Faculty members need a way to track project status and store project documents for retrieval.

Barriers

Cost is the top problem faced by an NGO in implementing technology. Second on the list of problems for an NGO is the time required to learn new technology and implement a technology solution. Finally, some NGO leaders cannot envision how technology may aid them to accomplish their missions. On the other hand, students may be limited to impractical and fictional problems posed in academic classes and are barred from tackling real problems because they may not have access to an organization which has technology needs.

XML solutions to be discussed

Each of these hurdles might be overcome if a student in computer technology were to team up with an NGO to create a project. To provide a web site that would unite the two, student and NGO, a document driven knowledge management system was proposed. A search for working XML projects that implement this type of system was conducted for this project. Five

potential solutions were found and researched for the Services Learning Practicum:

- Apache Cocoon
- Apache Lenya
- Microsoft Professional Office 2003
- Oracle 10g
- Orbeon

Table 2 XML Product Solutions

The proposed system also had to meet the business requirements outlined at the beginning of the project (Litz). Included in the requirements is the proposition that the XML solution proposed can manage the documents associated with projects by storing student works in a document repository for retrieval as needed by the faculty. Follow on projects can use the flexibility of XML to provide search algorithms to facilitate the ease of searching through all related documents.

Project Scope

Calendar requirements constrained the project to six months, November 2004 through April 2005. With one person working on the project, the amount of programming was limited to providing a rapid prototype of the system and not a fully functional software product. The project can be considered to be primarily a software design project. While prototype solutions were built and tested as part of the primary research, time was not allotted to fully implement a working solution that collected actual student or NGO documents for storage and retrieval. Implementation will need to be finalized in follow-on projects.

Definition of terms

Apache – web server solution and the name of the foundation that provides organizational, legal, and financial support to a broad range of open source products.

ERD - Entity Relationship Diagram is a diagram that represents the relationships between tables and records in a database

NGO – Non-Governmental Organization or Non-Profit Organization operating within the guidelines of US and International law

Orbeon – Orbeon is a service company that provides XML training and production support for their product, Presentation Server

XML – Extensible Markup Language, a W3C recommendation for text processing

W3C – The World Wide Web Consortium, a forum for information, commerce, and collective information that develops interoperable technologies

Summary

NGO leaders have technology needs and graduate students are in need of worthwhile projects that design, develop, and implement technology solutions. Faculty members need a way to manage projects. What better place for students to show altruistic leadership in advancing technology but by working on projects for non-profit organizations who need low cost or no cost technology to solve real world problems. The Student Learning Practicum provides a design for a knowledge management system that can track and manage these practical problems.

Chapter Two: Review of Literature / Research

Overview of literature and research on XML

The amount of literature which centers on XML design is staggering. Most information can be learned by research on the World Wide Web. Another very good source is the XML Handbook by Charles Goldfarb and Paul Prescod. Most of the design ideas for the project had their roots here or from the W3C recommendations (W3C). One of the business requirements of the project centered on the use of open source materials. "XML is a tremendous victory for open standards. It is freely extensible, imposing no limits on the ability of users to define markup in any combination of the world's major natural languages; it is character-based and human-readable, which means that XML documents can be maintained using even the most primitive text processing tools…"(Goldfarb liv). By using open source materials and sources, students and NGOs can keep the costs down and learn the open architecture that XML processing enables.

Project research

Primary gains from research for this project is outlined in the following table. In selecting and testing five XML engines the following chart shows which selection may be best for a project at hand.

Criteria	Cocoon	Lenya	Office 2003	Oracle 10g	Orbeon
Cost	Free	Free	$199 single InfoPath license ($199 for an academic license for Professional Office 2003)	$4.995 Perpetual standard edition license; $149 named user plus perpetual	Free
Knowledge	Deep Knowledge Required	Deep Knowledge Required	Smart tutorials available	Smart tutorials available	Deep Knowledge Required
Time to learn	Long term	Long term	Expedient	Expedient	Mid term
Usage	Common Web Use	Medium Use by some	Used often by business	Used often by business	Medium Use by some
Architecture	Open Source	Open Source	Proprietary	Proprietary	Open Source
Time to build	Long term	Long term	Expedient	Medium term	Long term

Table 3 XML Engine selection

"Apache Cocoon is a web development framework built around the concepts of separation of concerns and component-based web development" (Apache, Cocoon). Cocoon uses pipe-lines to glue components together in a "lego-like" structure.

Apache Lenya is an open source content management system built on Apache Cocoon and other components from the Apache software stack. Lenya is written in Java and is based on open source standards like XML and XLST (Apache, Lenya). The two Apache systems are worthwhile for server architects to study as web solutions on the server side of a complete XML knowledge management system.

Office 2003 is presented as an “off the shelf” XML solution. Because of Microsoft’s popularity as Office application software, this solution may be natural for many who currently use Microsoft products. The difference between Professional Office 2003 and previous versions is that now “using XML with Word documents enables companies to capture more of the intellectual property that is created informally to enterprise information systems. As XML, that property becomes a portable asset that can be reused as needed” (Goldfarb 418).

Around February of 1998, Oracle began to look seriously at the new and promising technology of XML. “In fact the Architectural Review Board decided that XML was going to be of company-wide importance and handed down a development charter to the CORE Development group as follows: *Deliver the best platform for developers to productively build and cost-effectively deploy reliable and scaleable Internet applications exploiting XML*” (Scardina xix). This statement best explains why Oracle was considered in the study for the Services Learning Practicum. The server product, Oracle Database 10g, was created with XML Internet applications in mind. It allows for a completely new data type called a “CLOB” or “Character Large Object.” A CLOB can store an entire large XML document within the Oracle database. “CLOBs can be indexed to search the XML as plain text or as document sections for more precise searches” (Scardina

11). Oracle brings a medium weight to those designing XML applications. If one currently uses an Oracle Database product, in order to convert from a relational database to the XML type of database one can do it with Oracle support. “Databases and XML offer complementary functionality for storing data. … Oracle 8i, Oracle 9i, and to an even larger extent, Oracle 10g enable you to store XML natively and build XML–enabled applications.

Orbeon is a service company that provides XML training and production support for their product, Presentation Server. In the Presentation Server Tutorial, an example application is called, “The Biz Doc Application (Orbeon). Biz Doc is a small portion of a fictitious business document processing application and allows the user to create, update, and delete insurance claims documents (Orbeon). What is useful in Orbeon is the examples are provided to modify to the type of document needed. In the case of the Services Learning Practicum, the documents to be processed are the Student Application and the NGO Needs statement. Another practical part of the Orbeon Company’s contribution is the Orbeon Studio. It is an open source development environment that provides a developer tools to create and refine XML documents and the XML pipeline language required to manage the document through a process like the Services Learning Practicum. A great deal of primary research through trial and error was conducted using the Orbeon studio.

Summary: what is known and unknown about XML

"Where data was once a mysterious binary blob, it has now become something ordinary people can read and author because it's text" (Goldfarb xlix). A great deal is made in the research about XML being both text and data. This is an essential part of the flexibility of XML and is what enables the Services Learning Practicum to manage documents, precisely because they are XML documents. "The computer world's excitement (about XML) can be summarized in two words: *information exchange*" (Goldfarb 5). This is what makes the choice of XML architecture for the Services Learning Practicum knowledge management system a natural fit. "XML is about making computer systems work together through the exchange of everything from simple numbers to elaborate data structures and human-readable texts" (Goldfarb 5). "XML is a framework for any project that involves moving information from place to place, even between different software products and platforms" (Goldfarb 6).

"The easiest way to understand the central ideas of XML is to go back to their source, the Standard Generalized Markup Language (SGML)" (Goldfarb 6). As a subset of SGML, like its sibling HTML, XML comes from a history of text processing systems. "Text processing is the sub-discipline of computer science dedicated to creating computer systems that can automate

parts of the document creation and publishing process" (Goldfarb 7). The file format saved by a text processor is called a *rendition*. "Some historically important rendition notations include *troff, Rich Text Format* (RTF – used in some Microsoft software), and *LaTeX* … newer programs like *Microsoft Word* and *Adobe Page Maker*[1] still work with renditions, but they give authors a nicer interface to manipulate them. The user interface to the rendition (the file with formatting codes in it) is designed to look like the presentation (the finished paper product). We call this *What you See is What you Get* (WYSISYG) publishing. Since a rendition merely describes a presentation, it makes sense for the user interface to reflect the end-product (Goldfarb 7).

What is unknown about XML is the extent to which it is being used in the world or how it will be used in the future. As an open-source recommendation from the W3C and the ability of one to compose XML on any text processor available makes it extremely popular as a web portal tool.

[1] Microsoft and Adobe (as well as Oracle) are large text processing software companies that have chosen to build software that utilizes XML technology.

Contributing aspects of this project

The research provided in the documentation of this project will aid others who are building XML portals for the purpose of knowledge management. The use of data processing for managing documents has progressed significantly in the last decade. With the use of mark-up languages for documents, such as hypertext markup and extensible markup, and the ability of a wide variety of people across the globe to access material posted on the World Wide Web has created nearly instant information on a nearly infinite variety of topics. The task to manage that information is increasingly dependent upon machine processing. In order to provide documents that can be both read by humans and processed by machines, the standard extensible markup language has been developed. It allows so much flexibility because it incorporates what is known as “metadata” right along with the human readable character data. For some the ability to create documents that contain what would normally be considered database schema type metadata make XML particularly useful for managing data content. In this project for instance, a contact can be a student, a faculty member, or a representative of an NGO. The metadata which surround the character data that represents the name provides a place to put that data much as rows and columns would place that data in a relational database. But within the XML schema, the paradigm is not rows and columns

but trees and branches. A natural XML document repository can store the same type of data as a relational database but XML provides for more efficient document retrieval. Search algorithms for the documents within an XML document repository search first for the metadata, then for the text allowing for quicker response times than traditional database searches. Search engine algorithms which have been built to find text on the Internet can also be applied to search through an XML document repository and provide detailed data as well as metadata which describe that data.

Chapter Three: Methodology

Research methods to be used

This research paper is a combination of primary research and secondary research. "Primary research is the study of a subject through first-hand observation and investigation. Secondary research is the examination of studies that other researchers have made on a subject" (Gibaldi 2).

The primary research method in this project was the design, construction, and testing of the prototype. In the design and construction phases the software code was actually manipulated and tested until it produced the desired results. This trial-and-error methodology provided great insight into both the flexibility and complexity of designing a web portal using XML.

The secondary research method was used in gathering design documentation and software tools in order to design and build the software for the prototype. Two sources provided the foundation to understanding XML; Charles Goldfarb's The XML Handbook (Goldfarb) and the XML recommendations of the World Wide Web Consortium (XML).

Life-cycle models to be followed

The rapid prototyping model was chosen for this project. Given the time constraints, the rapid prototype was chosen over other software life-cycles. “A rapid prototype is a working model functionally equivalent to a subset of the product” (Schach 70). The goal of this project was to build a software engine that could be tested, not merely to design one. To achieve that goal, a rapid prototype was created to show how XML can work to gather and store data. The figure which follows this text description shows the specific phases of the rapid prototype model. In software engineering one of the greatest challenges is change. At the beginning of a project requirements are assembled and specifications designed to meet those requirements. Developers have to accept and manage change in a project. A rapid prototype allows for nearly instant demonstration of a software product to the requirements analysts and business users. By selecting this type of software life-cycle, within the six months allotted, an actual software product was developed and demonstrated. The scope of the project did not pass beyond the line marked “Project duration” on the following figure.

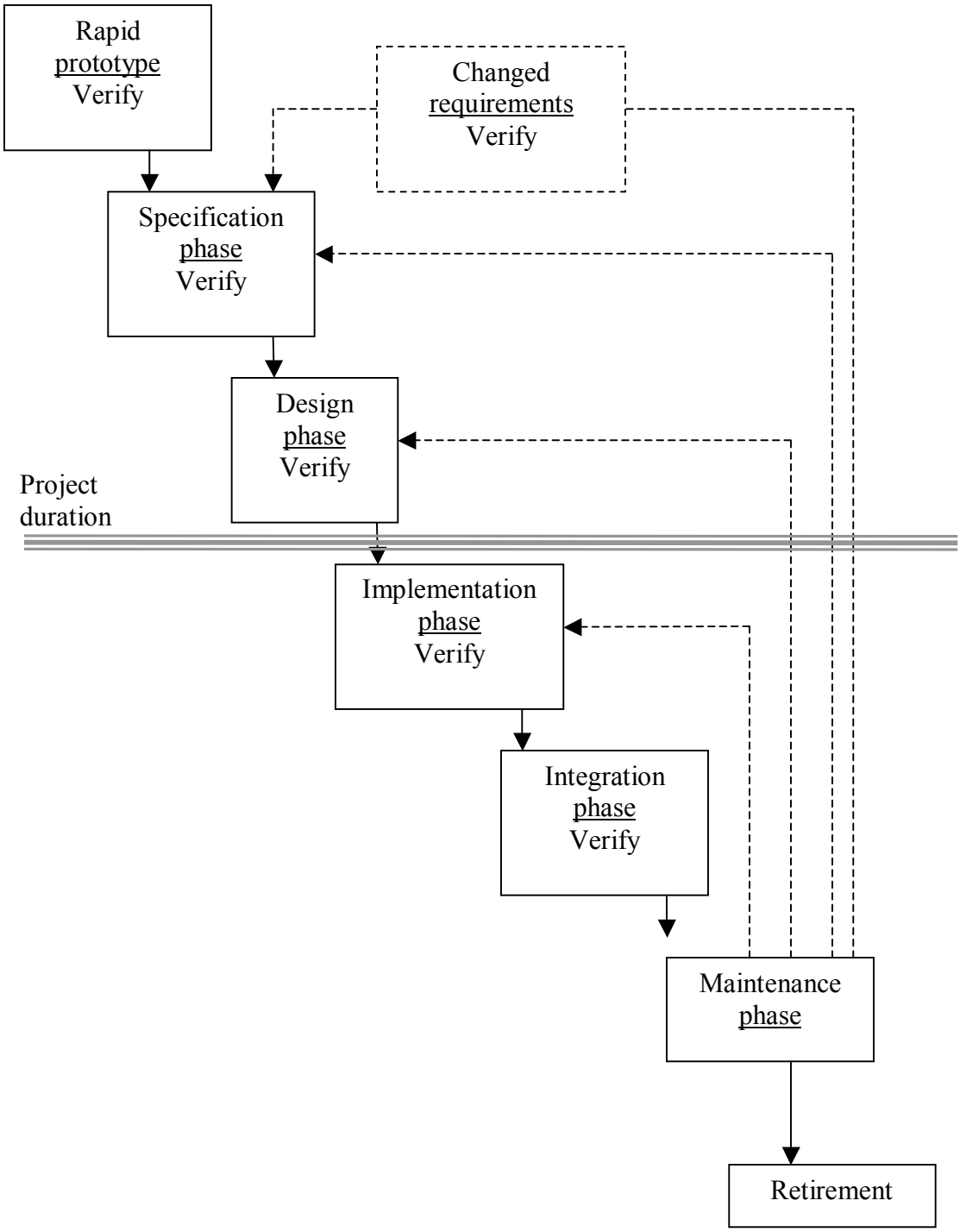

Figure 1 Rapid Prototype Diagram (Schach 71)

Specific procedures

The first prototype forms were written in XHTML using Amaya in November 2004 and posted to a prototype website: http://slp.regis.home.comcast.net/ Amaya is the W3C open source HTML and XHTML editor (Amaya). "Amaya is a Web client that acts both as a browser and as an authoring tool. It has been designed by W3C with the primary purpose of demonstrating new Web technologies in a What You See Is What You Get (WYSIWYG) environment. The current version implements the Hypertext Markup Language (HTML), Extensible Hypertext Markup Language (XHTML), Mathematical Markup Language (MathML), Scalable Vector Graphics (SVG), the animation module of Synchronized Multimedia (SMIL Animation), Cascading Style Sheets (CSS), and Hypertext Transfer Protocol (HTTP)" (Amaya). The forms for the Service Learning Practicum could be designed and built using a various open source and commercial products but Amaya was chosen because of its cost; Amaya is free and can be downloaded from the W3C website (Amaya). The figure that follows is a screen shot of the WYSIWYG editor holding a Student Application form designed for the Services Learning Practicum. The form was built and tested easily within minutes using this open source tool.

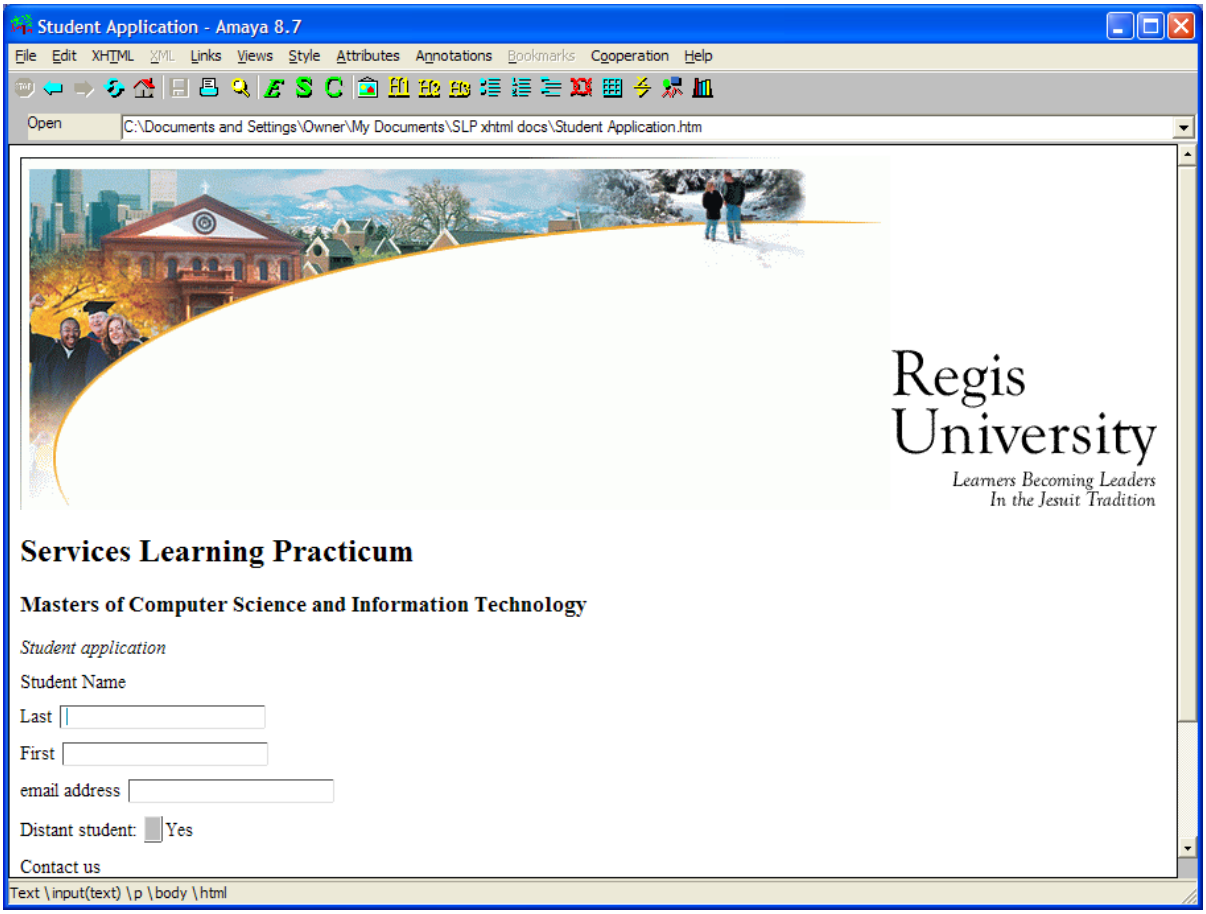

Figure 2 Amaya

Source code for the above form:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3c.org/TR/1999/REC-
html401-19991224/loose.dtd">
<!-- saved from url=(0055)http://home.comcast.net/~slp.regis/SLP/Application.html -->
<html>
<head>
  <title>Student Application</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<img alt="Regis University"
src="Student Application_files/RegisHomeImageSm01.jpg"><a
href="http://www.regis.edu/default.asp"><img
alt="Regis University, Learners becoming Leaders"
src="Student Application_files/logo.gif"></a>
<h1>Services Learning Practicum</h1>
<h3>Masters of Computer Science and Information Technology</h3>
<p><em>Student application</em></p>
<p>Student Name</p>
<p>Last <input type="text"></p>
<p>First <input type="text"></p>
<p>email address <input type="text"></p>
<p>Distant student: <button></button>Yes</p>
<p>Contact us</p>
<p></p> <p></p> <p></p> <p><code></code></p>
<SVG xmlns=http://www.w3.org/2000/svg version=1.0><TEXT x=177px
y=119px></TEXT></SVG></body>
</html>
```

Formats for presenting results/deliverables

Prototype documents were saved as HTML and XHTML documents and presented on the SLP website. The XML engines to drive the data collection were built on a notebook computer to simulate a web server and allow for prototype display to the review group.

Review of the deliverables

Meetings were scheduled with the faculty advisor and SLP group members to review the design and go over the form graphics and prototype. The first reviews were conducted in November and December 2004. Subsequently more research showed that while the initial proposal defined the project in terms of building a working model, the resources involved in actually running a website proved to be a constraint to completion of the second part of the project, the actual implementation of the website. However, the designs were proved and tested on a variety of platforms and were able to be displayed for review.

Resource requirements

Initially one programmer was available for writing the XHTML, HTML, and XML. One programmer was available to write the database and the XML schema. The second programmer left the project and this impacted the schedule. A Comcast web server

was available to host the prototypes. However, the XML document storage solution was not possible on this server because of restrictions placed on the server by the host. What was required was a host server which could house an XML engine like Apache Cocoon. Questions were addressed to find a host server for the project that would provide the necessary support. Other Regis Professional Studies Practicum web sites are hosted by commercial firms who provide maintenance support. No maintenance support can be provided on Regis servers. A plan was launched to start the project running on a web server late in the project development stage; however the project never made it out of the prototype phase into the implementation phase. The next generation of this project could move on with clear focus to the Internet to collect student and NGO data.

Outcomes

The rapid prototype life cycle produced a product that could be judged and new requirements were generated from the demonstration. The prototype proved to be an immediate visual demonstration of the project's potential.

Using open source tools proved to be frustrating and yet the "no cost" and availability of these tools to anyone is a big plus. Examination of the more expedient Office 2003 and Oracle 10g solutions proved that this sort of XML document storage is

new and promising technology which is being embraced by large companies like Microsoft, Adobe, and Oracle.

Summary

The final prototype was built and functioned as expected using Microsoft Office 2003. The schema built using open source Orbeon Studio development environment was placed into the Microsoft InfoPath application which is part of Professional Office 2003.

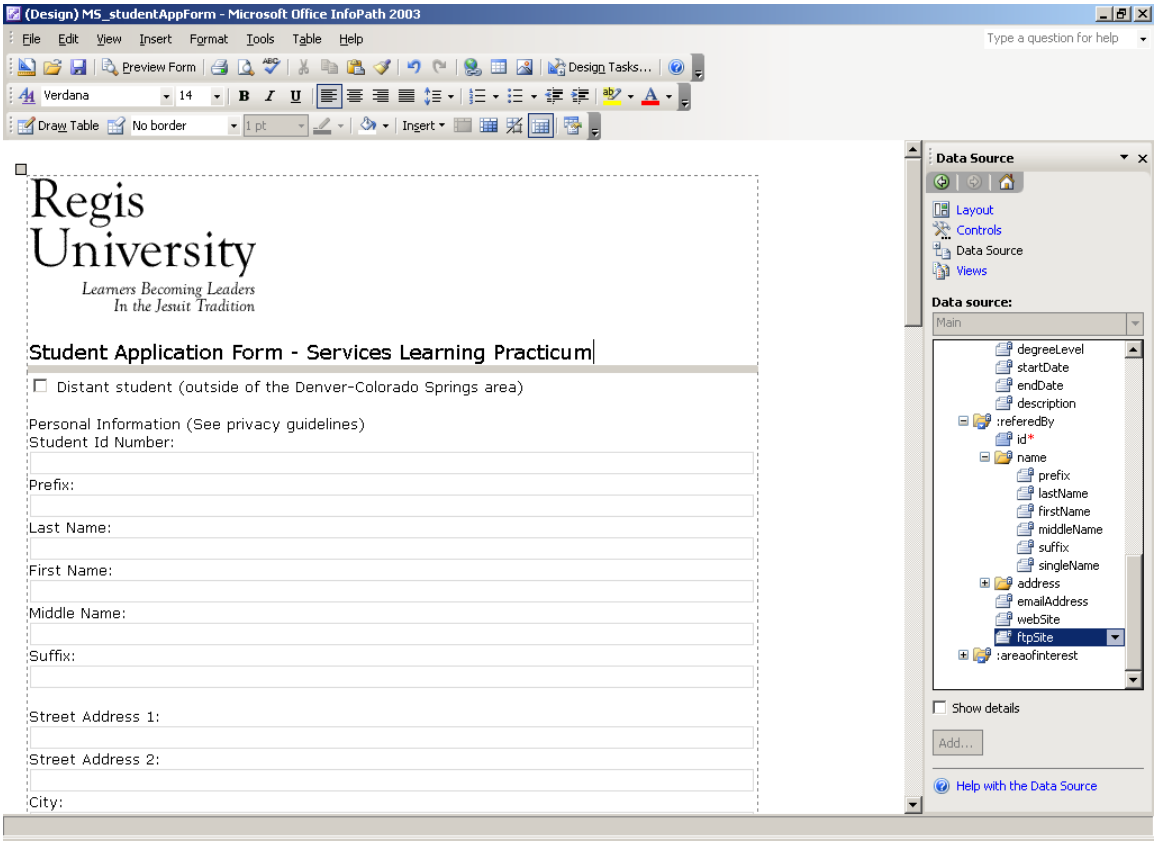

Figure 3 InfoPath

The InfoPath form editor is a “What you see is what you get” type of editor that displays the form in the format one would be able to see in a web browser. People are becoming more and more familiar with this sort of word processing. This document, itself, was created using Microsoft Word which has become a standard word processing application. Forms built with InfoPath also allow dynamic entry. “Common productivity tools like office suites supported on specific XML document types, when they supported XML at all. Microsoft Office Professional Edition 2003 changed that situation forever by accepting custom schemas as first-class citizens. As a result, millions of desktop computers were transformed from mere word processors into rich clients for Web services, editors for XML content management systems, and portals for XML-based application integration” (Goldfarb 415).

The data entry forms used by the Services Learning Practicum can be built using open source software and detailed programming. However, as a demonstration, the Service Learning Practicum Student Application pictured above was built in a short thirty minute session. Some InfoPath features that might be familiar to Web developers are the use of style sheets. “Unlike Word, InfoPath generates an XSLT style sheet to control the rendering of the form, so there is no ‘InfoPathML.’ The formatting can even be based on data entered in the form”

(Goldfarb 419) For example, a negative value on a ledger data entry form might be in a different colored text, like red for negative numbers.

So once the data is captured in a form, what happens next? XML provides the pathway language and the means to process it. “XML elements, whether captured in Word or Excel or InfoPath are as well-defined and predictable as the columns and tables of a database” (Goldfarb 420). (Construction of XML does not depend on the tool such as Word, Excel, or InfoPath used to create it; if time permitted, one could labor over an XML document solution using a simple text editor) This was proven with the unit tests of the schema defined for SLP in Microsoft Development Environment (.Net). One can review the results of that testing in Addendum B of this document.

Chapter Four: Project History

Project Origin

The Services Learning Practicum began with the research and design of Joyce Snowden who developed the Professional Project Workbook and Workflow as a Master's Degree Project at Regis University (Snowden). Snowden's lead project identified a process to track and store project information for students engaged in Master's Degree Professional Projects.

The Services Learning Practicum was launched to meet the needs of two different groups; graduate students looking for practical application of their technical and professional knowledge and non-profit organizations looking for inexpensive technology solutions. The three objectives of the project became 1) to meet the needs of students, 2) to meet the needs of non-profit organizations, and 3) to track, manage and store the knowledge and project papers generated by the Services Learning Practicum.

Project Management

The project itself was managed as an academic research project. As such, the project fell naturally into five phases; 1) business requirements phase, 2) technical specifications phase, 3) project proposal phase 4) prototype development phase,

and 5) preparation of academic documents and presentation phase. The rapid prototype software life cycle was used to manage software development and testing. Defects uncovered during testing were fixed and the software retested.

Project Milestones

Phase 1, November 2004, the business requirements and database schema were discussed and proposed for the project.

Phase 2, November 2004, the business requirements were analyzed and developed into a specification document (Addendum A).

Phase 3, December 2004, the initial proposal and data schema were finalized and presented to the faculty advisor and other project developers for review (Addendum B).

Phase 4, March 2005, final prototype designs were presented for demonstration and review.

Phase 5, April 2005, academic papers and presentation were made available for faculty and student reviews.

Changes to the project plan

The changes in the initial project plan reduced the scope of the project due to lack of resources working to achieve the goal of implementing a prototype and collecting documents for the repository by April. The first concept of the project set

the expectation based upon the assumption that a server at the University might be made available to host the practicum. In the course of investigation the discovery was made that other professional project practicum websites are hosted by service providers that provide server maintenance at a cost to the project. This was not known at the beginning of the project and reduced the scope to a prototype development project.

Additionally, a student database programmer was also initially available for the project but was called on for other work and was not available to build the XML schema for the project. That left one programmer to complete all programming tasks for the project.

Evaluation of project goals

The requirements analysis, specification, design, and rapid prototype goals of the project were met. However the implementation goal to collect actual data from students and non-profit organizations has not yet been realized. Other goals included study and a gain in knowledge about XML and how to implement a knowledge management document driven system on the Web. The knowledge gained in this technology that is rapidly gaining popularity with web designers was worth all the effort to complete the project within the short time allotted.

Discussion: Project achievement and frustrations

The initial scope of the project was over estimated. This is due to a lack of personnel resources to carry out all the programming tasks and a lack of a host server.

The Services Learning Practicum as an academic research project was a great success. The project documents contain instruction and actual source code on how to build a knowledge management system using XML. Several XML engines and tools are described in detail that are available and would be valuable to one who is beginning research into knowledge management with XML documents.

Discussion: Impact of project variables

1) Host server; where to host the website was a big concern at the onset of the project. In November, the project team discussed where the site would be hosted and agreement was made to have the prototype completed in March to begin document submission and storage by April 2005.
2) Schema design and development; the original database design and ERD (Addendum B) created by Joyce Snowden were prepared for an Oracle relational database. An XML document repository was selected as better suited for

the Services Learning Practicum architecture and an XML schema (Addendum B) was created for the project.

3) XML Engine; which XML Engine to choose for the prototype was a major part of the research and analysis. Five candidates were selected for review and are displayed in Table 2, Chapter 2.

Findings/analysis

1) XML can be used effectively for building a document-driven knowledge management system.
2) Open source XML tools while inexpensive are time consuming to use.
3) Open source XML tools require those using the tools to learn a medium amount of mark-up language skills to be able to build XML documents.
4) WYSIWYG editors like InfoPath and Amaya allow students with minimal programming experience to develop source code for a project.

Chapter Five: Lessons Learned / Next Evolution

The project experience

By organizing the work into a project, the goals and requirements could be better tracked and managed. Of great benefit was to set the scope of the project early on so as to define the direction the development effort needed to take and the timeframe in which to accomplish all the required tasks.

Without the framework the project provided, the Services Learning Practicum study would not have been successful at all. Without a proposed calendar of action and without clearly defined requirements, the overall software life-cycle plan could not have been realized.

What could have been done differently

The academic and secondary research goals should have been addressed first as the top priority; then the construction of the prototype should have been the secondary goal after the secondary research was completed. Both were done simultaneously and as a result, the boundaries between primary and secondary research were blurred. A much better approach would have been to complete the necessary academic paper which would have included the secondary research on XML as it applied to the

Services Learning Practicum design. Once completed, the primary research of actually building and testing a prototype would have probably been more fruitful.

This academic project as part of Master's degree program should have been limited to an individual student so as to demonstrate single-minded achievements in both academic study and the application of computer science. Initially in this project there was a sense that more students would be participating. This created assumptions that more work could be done. While collaboration is good, especially given the nature of the computer science industry, a case can be made that an academic research can best be conducted by an individual student. Because of this assumption, the schedule and tasks for the project needed to be reallocated and the scope reduced to get the project completed on time.

Discussion: achievement of initial project expectations

The initial project expectations were to have a website collecting data from students and NGOs before April 2005. While not unrealistic, the initial goals were not met because of two separate issues:

1) The number of student's programming the software for the prototype was reduced from two to one.

2) A server could not be procured in time to host the Services Learning Practicum XML engine or XML portal.

Next stage of project evolution

Follow-on projects can benefit by this project in that the Services Learning Practicum project papers discuss the open source materials available for XML design and knowledge management. Another project which follows this one could actually place the XML portal on a website to begin data collection. The forms that would drive this collection would be the Student Application and the NGO Needs Statement XML documents.

Another stage to the project might be the introduction of the document repository for project documents. Student papers, presentations, and web designs could be placed in this repository for future reference.

Further research could be conducted on development of search algorithms or search engines to find and display the information placed in academic project papers stored in the Services Learning Practicum document repository.

More work is also needed on defining and creating the pipe-line language to manage the knowledge stored in the system. This work can be done as more and more documents have been included in the system.

Conclusions/recommendations

A document driven knowledge management system can be built to manage the information needed to unite graduate students with NGOs for meaningful and worthwhile computer science projects. XML has proven to be a flexible and useful means to achieve the Services Learning Practicum goals for two primary reasons:
1) XML is becoming widespread as a means to manage documents on the World Wide Web. “All of the major software companies are enthusiastic about XML. New industry standards based upon it are released daily” (Goldfarb 5).
2) XML is an open source recommendation of the World Wide Web Consortium (W3C) and therefore can be implemented with little or no cost to students or NGOs.

If time is not an issue, the best choice for complete application development and hosting is the solution presented by Orbeon. The Orbeon Presentation server comes with a development studio and plenty of examples that will lead the XML developer to success in building a fully operational system including Graphical User Interface and server components for document repository. The learning benefit in using Orbeon is valuable. One will learn many of the fundamental parts of the XML language and especially how to design XML pipe-lines.

If time is limited but funds available, utilizing Microsoft Professional Office 2003, specifically InfoPath, would prove to

be a workable solution for many. While moderately expensive, the benefit of needing less time for training and development might outweigh the overall costs. Future Microsoft Office versions will probably include features available in Office 2003 and therefore one might expect that in the future, XML document solutions created with InfoPath, Excel, or MS Word 2003 will be as common as MS Word documents and Excel spreadsheets are today.

Summary

This project has come down two separate paths. On the one hand, this project is about the business side of managing projects for graduate students and NGOs. This effort feeds into some of the values that Regis University espouses; namely that learners become leaders. In this case, the graduate students may benefit in the future in that the work that they do may have practical application for those in need of their work.

On the other hand, this has been a technical research and learning program that has led to a technical understanding of the potential of XML in developing a knowledge management system. The flexibility and versatility of XML provide ways to manage documents that are no doubt being utilized today and will be used in the future. So learning XML has been a useful exercise. The knowledge gained in learning about XML data

structure and XML pipeline language will no doubt prove useful, to those who seek to study it. The hope is that the practical application of the research done for this project can be applied to future projects.

Knowledge management has long been done by those in academic and research fields. One usually considers going to a library to conduct that research. Today, with so much information available online, managing the tremendous amount of information has become a sizable task. What better way to manage it than with eXtensible Markup Language.

Historically, one of the most interesting discoveries recaptured in this project is the origin of markup. The world's libraries could be considered the storehouse of human knowledge. And the art of markup can be traced to early printers who marked manuscripts to prepare them for printing. Today, the markup becomes "metadata" or data above the data which can be used for machine processing the documents. Metadata can provide different renditions of the same data for different users, thus providing the kind of personalization required for a project like the Services Learning Practicum. For instance, while an NGO may see "database development" as a need, a graduate student may see that character data as the area of interest for a project. Uniting the two by presenting the same data to each of

them in separate ways is the essence of the flexibility of XML in the project.

So this project has seen the potential of taking a historical method of managing knowledge, research at a library, and provided a vision of managing knowledge using XML documents. The progress down this pathway is not only logical but is proven to be the pathway taken by large software companies like Microsoft and Oracle. Each of them see the future in XML processing and storage. It only makes sense to use the latest technology to solve some of the more practical problems faced by computer scientists of today.

Works Cited

Amaya. Amaya Page. 21 Oct. 2004. W3C. 27 Oct. 2004. http://www.w3.org/Amaya/

Apache, Cocoon Project. 27 Dec. 2004. The Apache Software Foundation. 28 Dec. 2004. http://cocoon.apache.org/

Apache, Lenya. 09 Dec. 2004. The Apache Software Foundation. 28 Dec. 2004. http://lenya.apache.org/

Apache, License Version 2.0. Welcome The Apache Software Foundation. 26 Mar. 2005. http://apache.org/

Garfinkel, Simson and Gene Spafford. Web Security, Privacy, and Commerce. Sebastopol, CA: O'Reiley & Associates, Inc., 2002.

Gibaldi, Joseph. MLA Handbook for Writers of Research Papers. New York: Modern Language Association of America, 2000.

Goldfarb, Charles F. and Paul Prescod. XML Handbook 5th Edition. Upper Saddle River, NJ 07458: Prentice Hall PTR, 2004.

Goodyear, Mark. Enterprise System Architectures Building Client / Server and Web-based Systems. Boca Raton: Auerbach Publications, 2000.

Jakarta, Tomcat. The Jarkarta Site. 27 Dec. 2004. The Apache Software Foundation. 28 Dec. 2004. http://jakarta.apache.org/tomcat/

Jigsaw, W3C's Java Web Server. Jigsaw Overview. 18 Nov. 2004. W3C. 18 Mar. 2005. http://www.w3.org/Jigsaw/.

Litz, Trisha. Service Learning Practicum Initial Business Requirements, MS Word document, Regis University, November 10, 2004

Orbeon, 12 Dec. 2004. Orbeon. 1 Jan. 2005. http://www.orbeon.com/

Orbeon, “Orbeon Presentation Server Tutorial, An Introduction to the Orbeon Presentation Server” Orbeon, Inc., 2004, http://www.orbeon.com/

Scardina, Mark, with Ben Chang and Jinyu Wang. Oracle Database 10g XML &SQL: Design, Build, & Manage XML Applications in Java, C, C++, & PL/SQL, Oracle Press, 2004

Snowden, Joyce. “MSCIT Professional Project Workbook and Workflow,” Regis University, Master of Science in Computer and Information Technology, 2003

XML, Extensible Markup Language. 27 Dec. 2004. W3C. 28 Dec. 2004. http://www.w3.org/XML/

W3C, World Wide Web Consortium. 17 Mar. 2005. W3C. 18 Mar. 2005. http://www.w3.org

W3C Schema Primer, 28 Oct. 2004. W3C. 6 Dec. 2004. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/primer.html

Addendum A: Business Implementation

Implementation of Business Requirements using Extensible Markup Language (XML)

Document type: Software Requirement Specification (SRS)

Business Requirements

Purpose: Business Requirements are specified in this document. The business requirements defined by Professor Litz are mapped to software requirements for the SLP portal. The term "portal" will be used throughout this document to describe the net centric computer system that interacts with various users. The term "administrator" may refer to a human administrator who can access the entire system and make changes or to an automatic administrator that can perform automated changes within the portal like adding documents to repositories upon submission or review.

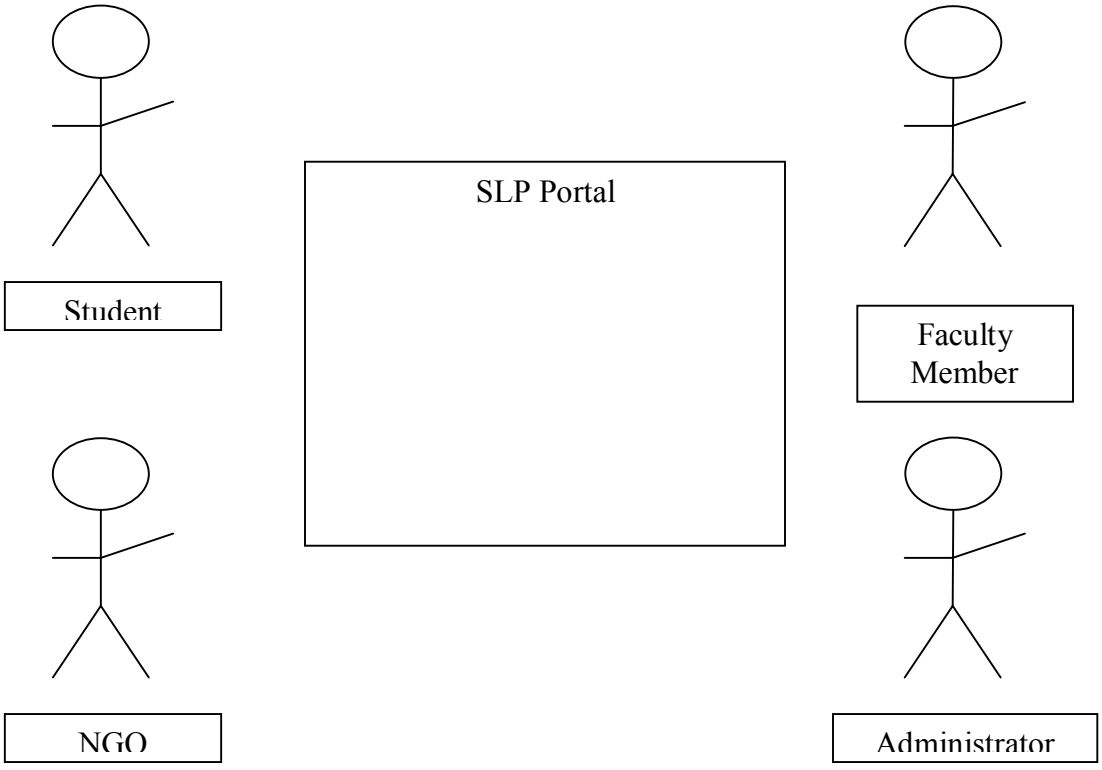

Figure 4 Business Use Case Diagram

1. “Central Web site for students, faculty, Non-Governmental Organizations (NGOs) and administrators to use”
    a. The SLP must be a net centric portal from the World Wide Web (Internet).
    b. The initial security level must require registration by the user as student, faculty, NGO, or administrator.
    c. The hyperlink path that the user will follow must be unique to the user’s level of access as student, faculty, NGO, or administrator.
    d. The portal must be personalized for each user.
    e. The portal must be secured with a user login protocol.
    f. The portal must provide a general access index page that allows user login and provides high level information about the SLP.
2. “Students can find out information about what the SLP is and what types of projects are done in the SLP”
    a. Web pages accessed by the student must include information documents that provide a high level of organization and detail as to the projects sponsored by the SLP called “Project Papers”
    b. Students have access to a specified set of “Project Papers”
    c. NGOs have access to a specific set of “Project Papers”
    d. Faculty and administrators must have access to all “Project Papers”
3. “Students can apply to the SLP online”
    a. Students must register through the portal or with a written “Student Application” which is then entered by Regis University staff.

   b. Students who submit an online application will create an XML document called “Student Application” with the root element name “Application”
   c. Data entered by the student must be checked for data entry errors against the Application Schema
   d. Students must be prompted immediately if data entry errors prohibit the publication of the Application. If the student fails to correct the data errors, the portal will notify an administrator and only a partial application will be posted.
   e. Minimum requirements for a partial Application and limited access to the student portal are the student name, email-address, country, postal code, and at least one phone number.
4. “Faculty can review and accept student applications online”
   a. Once a completed and verified Application (data checked against the Application Schema) has been submitted to SLP, the portal will notify an administrator and store the Application in a queue for review by faculty.
   b. Once marked “Reviewed” by a faculty member the Application must be stored in the Student Application Repository and be available for data retrieval.
5. “Projects can be created and managed online”
   a. A “Project” is an XML document that unites a student with an NGO to accomplish a specified task to meet the “Need” of an NGO.
   b. A “Need” is defined by the NGO in the “Needs Statement” document.
   c. Only a faculty or administrator can create a “Project”
   d. Only a faculty member or administrator can assign or reassign a “Project.”

   e. The portal must display in a concise manner “Needs” that match student candidate fields of “interest”, “experience”, “knowledge”, and “coursework” taken from the “Application.”

6. “Students can read background information for their assigned project accessed through the Web site”
   a. Once assigned to a “Project” the portal will notify the student upon login and provide a link to a personalized portal for the student which includes other links to project specific documents such as the “Project Calendar.”
   b. The personalized portal may be configured by the student to include links, contact information, and access to update information like name changes, address or phone changes.
7. “Students are tested on the basics of their assigned project online”
   a. The faculty member listed as the “Reviewer” must provide criteria by which the project is tested.
   b. The student must meet the criteria.
8. “Students and faculty can submit, track and edit project deliverables and time frames online”
   a. The portal will create a “Project Calendar” for deliverables
   b. The “Project Calendar” must be an XML document with the element name “Calendar.”
   c. The student and the faculty member will create and update items on the “Project Calendar”.
9. “NGOs can submit application to be a part of the SLP online”

a. NGOs must register with through the portal on with a written “Needs Statement” which will be entered by Regis University staff.
b. NGOs who register online will create an XML document called “Needs Statement” with the root element name “Statement”

10. Students can be matched with NGO projects based on the students’ skills and interests and the NGO project needs, goals, deliverables.
    a. The portal must display student candidates and NGO needs to the faculty members during the review process and the project creation process.
    b. See “Project creation and Management” above
11. Key user can search the information on the SLP site to pull up different types of documents/information. For example, if I want to know all the NGOs that are requesting e-commerce functionality, I should be able to pull that information up.
    a. The “Statement,” “Application”, and “Project” repositories must be searchable.
    b. Access to document retrieval is controlled by user security levels.
12. “Perform research on project information, deliverables to write white papers, etc. for publishing and presenting at conferences and seminars and to obtain financing for grants/foundations or to fulfill grant obligations.”
    a. Each XML document stored in the repositories must have been verified against a published schema.
    b. The schema metadata will allow for reuse of XML document content for portal display, printing, or other means of publication.

13. "Track the different SLP students on their Professional Project courses"
    a. see Manage Projects and Test Students above
    b. The portal will provide a personalized web page for faculty members.

Document Service

In order to capture the XML documents inside the portal a service needs to be used to collect, verify, and organize the documents for storage and presentation. Several open-source engines to accomplish this task already exist. Work will need to be done on the server side to provide the service inside the SLP Portal. A great deal of my research paper will compare and contrast three open source portal engines.

- Apache Cocoon
- Apache Lenya
- Orbeon

All three use the Model, View, and Controller (MVC) Architecture.

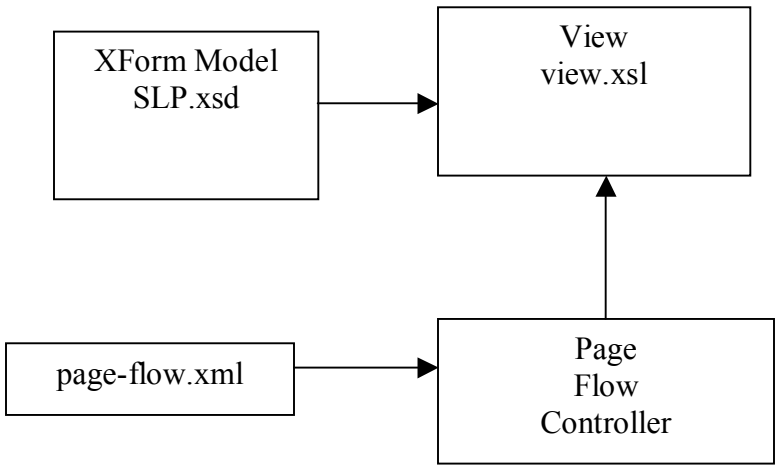

Figure 5 MVC Architecture

The “view” is the graphic user interface that is displayed, but behind the scenes in the server a model or schema is used to control the data (for example making sure that only numbers are entered into the phone number field). In addition to the model, the page flow controls what happens when for example, the “submit” button is pressed on a form. The SLP project depends on the choice of which engine to use. At a minimum the SLP project will be dependent on a service resident on a web server to accomplish collection of XML applications and needs statements.

Addendum B: XML Schema

XML Schema for Services Learning Practicum

An Entity Relationship Diagram (ERD) is "a diagram that represents the relationships between tables in the database" (Snowden, 27).

The ERD presented by Joyce Snowden in her paper is as follows:

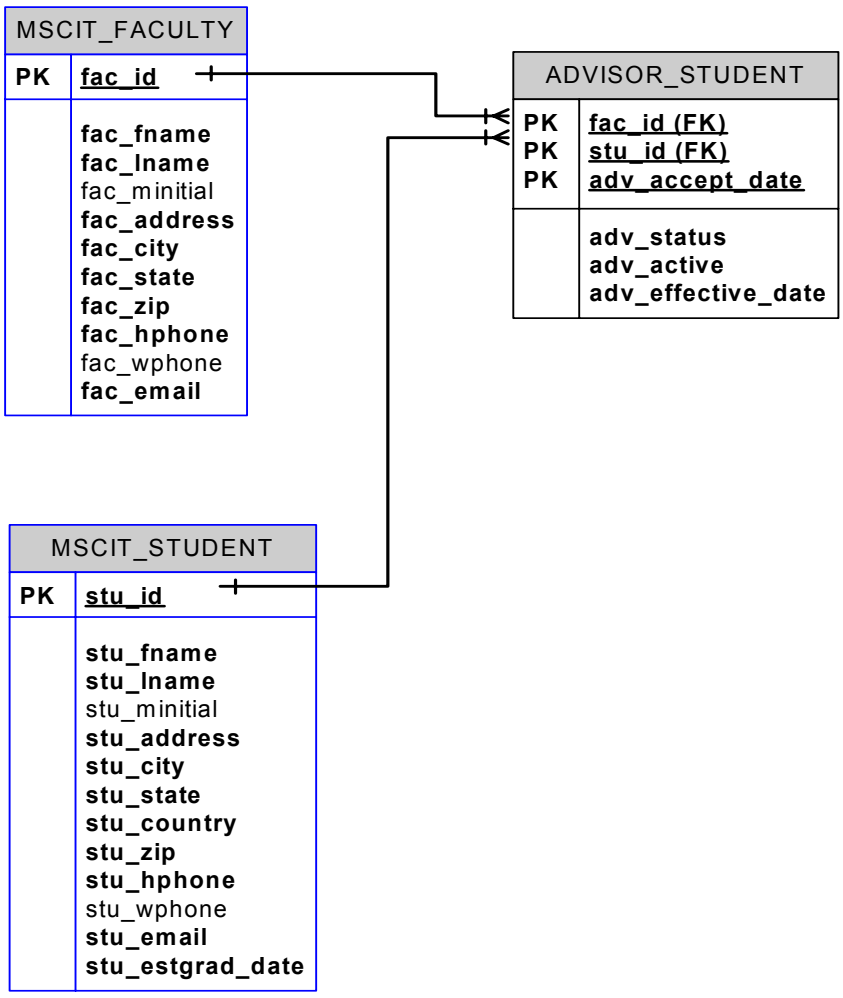

Figure 6 Workbook, workflow ERD

To convert the ERD from a relational database design into the XML schema one can create a namespace called “slpns” for example and use the namespace to define the schema. Using this model, the XML schema can be illustrated on the following pages (note that the address element is a child of the faculty element and can be defined further as an attribute itself; this demonstrates the power and flexibility of the XML schema):

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:slp="http://slp.regis.home.comcast.net/SLP/slpns"
     target Namespace="http://slp.regis.home.comcast.net/SLP/slpns">
     <xsd:element name="faculty">
      <xsd:complexType>
          <xsd:sequence>
               <xsd:element name="lastname" type="xsd:string"/>
               <xsd:element name="firstname" type="xsd:string"/>
               <xsd:element name="middlename_initial" type="xsd:string"/>
          </xsd:sequence>
          <xsd:element name="address">
             <xsd:complexType>
             <xsd:sequence>
                 <xsd:element ref="slpns:street_address_1" />
                 <xsd:element ref="slpns:street_address_2" />
                 <xsd:element ref=" slpns:city" />
                 <xsd:element ref=" slpns:state" />
                 <xsd:element ref=" slpns:zipcode"/>
                 <xsd:element ref=" slpns:homephone"/>
                 <xsd:element ref=" slpns:workphone"/>
                 <xsd:element ref=" slpns:emailaddress"/>
            </xsd:sequence>
            <xsd:attribute name="id" type xsd:ID"/>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="facultyaddress" type="slpns:address"/>
      </xsd:complexType>
     </xsd:element>
```

And so forth....

Another security consideration in the design was noted during project development. In order to collect and store information about a person on the Internet, one needs to be considerate of privacy laws and customs, especially as they pertain to storing the private information about individuals. In MSCE-610 E-Security, as part of the research associated with this project, information about identity theft was found that might prove useful in this and follow-on projects:

Technology is lagging behind the thieves in identity theft. One tool that is beginning to be used more widely is P3P. The World Wide Web Consortium's Platform for Privacy Preferences Project, (P3P) provides a standard for web sites to communicate practices about collection, use, and distribution of personal information. P3P uses XML expressions to define its policies. All P3P policies are downloaded from the web site itself. Because a web site's P3P statements may be intimately related to a web site's written privacy policy, an organization that treats personal information in a manner that is inconsistent with its P3P statements may be guilty of committing an "unfair trade practice" and may be opening itself up to an enforcement action by the Federal Trade Commission (Garfinkel 607).

Some questions raised about privacy are very important to answer in regards to Joyce Snowden's proposed schema. For instance, why do we need to gather and store a student's mailing address?

The purpose of storing contact information like email address, web site, and phone numbers seems necessary. Information on how to communicate would be required for transmissions between the student and the faculty and between the student and the NGO representative. On the other hand, a mailing address would not be necessary and might compromise security and privacy. If a mailing address would be required, one of the individuals within the practicum would be able to contact another individual for that more sensitive information.

The schema was created with the mailing address fields, however, they might be considered to be unimportant, optional, or just take them out of the forms altogether. The final schema which was built with the "optional" data included was inserted into Microsoft InfoPath and Microsoft .Net development environment to produce forms and an ERD.

Here is the finalized XML schema script:

```
<!--
   Copyright (C) 2005 Regis University, Services Learning Practicum (SLP).
   Author: Andrew C. Swanson Last modified 24 FEB 2005
   Faculty Advisor: Trisha Litz

   This XML schema encorporates the two driving documents of the SLP; the student
   applicaiton and the NGO needs statement.  The combination of the two create an
   SLP project.  The key fields that match a student to an NGO is the student
   areaofinterest and the NGO areaofneed.

   This program is free software; you can redistribute it and/or modify it under
   the terms of the GNU Lesser General Public License as published by the Free
   Software Foundation; either version 2.1 of the License, or (at your option)
   any later version.

   This program is distributed in the hope that it will be useful, but
   WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
   or FITNESS FOR A PARTICULAR PURPOSE.
   See the GNU Lesser General Public License for more details.

   The full text of the license is available at
   http://www.gnu.org/copyleft/lesser.html
-->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://slp.regis.edu/2005/slpns"
        xmlns:slp="http://slp.regis.edu/2005/slpns">
        <xsd:element name="student">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="personalInformation" type="slp:contactType" />
                                <xsd:element name="objective" type="slp:blockType" />
                                <xsd:element name="workExperience" type="slp:blockType" />
                                <xsd:element name="education">
                                        <xsd:complexType>
                                          <xsd:sequence>
                                                <xsd:element name="school" maxOccurs="unbounded">
                                                        <xsd:complexType>
                                                                <xsd:sequence>
                                                <xsd:element name="schoolName" type="xsd:string" />
                                                <xsd:element name="location" type="xsd:string" />
                                                <xsd:element name="degreeLevel" type="xsd:string" />
                                                <xsd:element name="startDate" type="xsd:string" />
                                                <xsd:element name="endDate" type="xsd:string" />
                                                <xsd:element name="description" type="slp:xhtml" />
                                                                        </xsd:sequence>
                                                                </xsd:complexType>
                                                        </xsd:element>
                                                </xsd:sequence>
                                        <xsd:attribute name="label" type="xsd:string" use="required" />
                                        </xsd:complexType>
                                </xsd:element>
                        <xsd:element name="referedBy" type="slp:contactType" maxOccurs="unbounded" />
                                <xsd:element name="areaofinterest" maxOccurs="unbounded">
                                        <xsd:complexType>
                                                <xsd:attribute name="yearsExperience" type="xsd:string" />
                                                <xsd:attribute name="courseHours" type="xsd:string" />
```

```
                                </xsd:complexType>
                        <!-- The area of NGO need matches up with the student areaofinterest field -->
                        </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="distant" type="xsd:boolean" />
        </xsd:complexType>
</xsd:element>
<xsd:element name="NGO">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="NGOContact" type="slp:contactType" />
                        <xsd:element name="needs" type="slp:blockType" />
                        <xsd:element name="dateSubmitted" type="date" />
                        <xsd:element name="dateEarliestDeadline" type="date" />
        <xsd:element name="otherRegisPersons" type="slp:contactType" maxOccurs="unbounded" />
                <!-- otherRegisPersons can be contacts or groups or referedBy contact information -->
                        <xsd:element name="areaofneed" maxOccurs="unbounded" />
                <!-- The area of NGO areaofneed matches up with the student areaofinterest field -->
                </xsd:sequence>
        </xsd:complexType>
</xsd:element>
<xsd:complexType name="blockType">
        <xsd:sequence>
                <xsd:element name="description" type="slp:blockXhtml" maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:attribute name="label" type="xsd:string" use="required" />
</xsd:complexType>
<xsd:complexType name="blockXhtml" mixed="true">
        <xsd:sequence>
                <xsd:any namespace="http://www.w3.org/1999/xhtml" processContents="lax"
                minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="addressType">
        <xsd:sequence>
                <xsd:element ref="streetAddress1" type="xsd:string" />
                <xsd:element ref="streetAddress2" type="xsd:string" />
                <xsd:element ref="city" type="xsd:string" />
                <xsd:element ref="state" type="xsd:string" />
                <xsd:element ref="zipcode" type="xsd:string" />
                <xsd:element ref="timeZone" type="xsd:string" />
                <xsd:element ref="phone1" type="xsd:string" />
                <xsd:element ref="phone2" type="xsd:string" />
        </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="contactType">
        <xsd:sequence>
                <xsd:element name="name" type="slp:nameType" minOccurs="0" />
                <xsd:element name="address" type="slp:addressType" minOccurs="0" />
                <xsd:element ref="emailAddress" type="xsd:string" minOccurs="0" />
                <xsd:element name="webSite" type="xsd:anyURI" minOccurs="0" />
                <xsd:element name="ftpSite" type="xsd:anyURI" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:ID" />
</xsd:complexType>
<xsd:complexType name="nameType">
        <xsd:sequence>
```

```
                    <xsd:element name="prefix" type="xsd:string" minOccurs="0" />
                    <xsd:element name="lastName" type="xsd:string" minOccurs="0" />
                    <xsd:element name="firstName" type="xsd:string" minOccurs="0" />
                    <xsd:element name="middleName" type="xsd:string" minOccurs="0" />
                    <xsd:element name="suffix" type="xsd:string" minOccurs="0" />
                    <xsd:element name="singleName" type="xsd:string" />
                </xsd:sequence>
            </xsd:complexType>
</xsd:schema>
```

Figure 7 SLP Schema

The schema was built using an open source development environment used to develop presentation server objects called Orbeon Eclipse Studio. The schema text was then loaded into the Microsoft Development Environment ".Net" and the schema was immediately displayed in the more familiar ERD format. This unit test verified the validity and applicability of this schema to be used in further development.

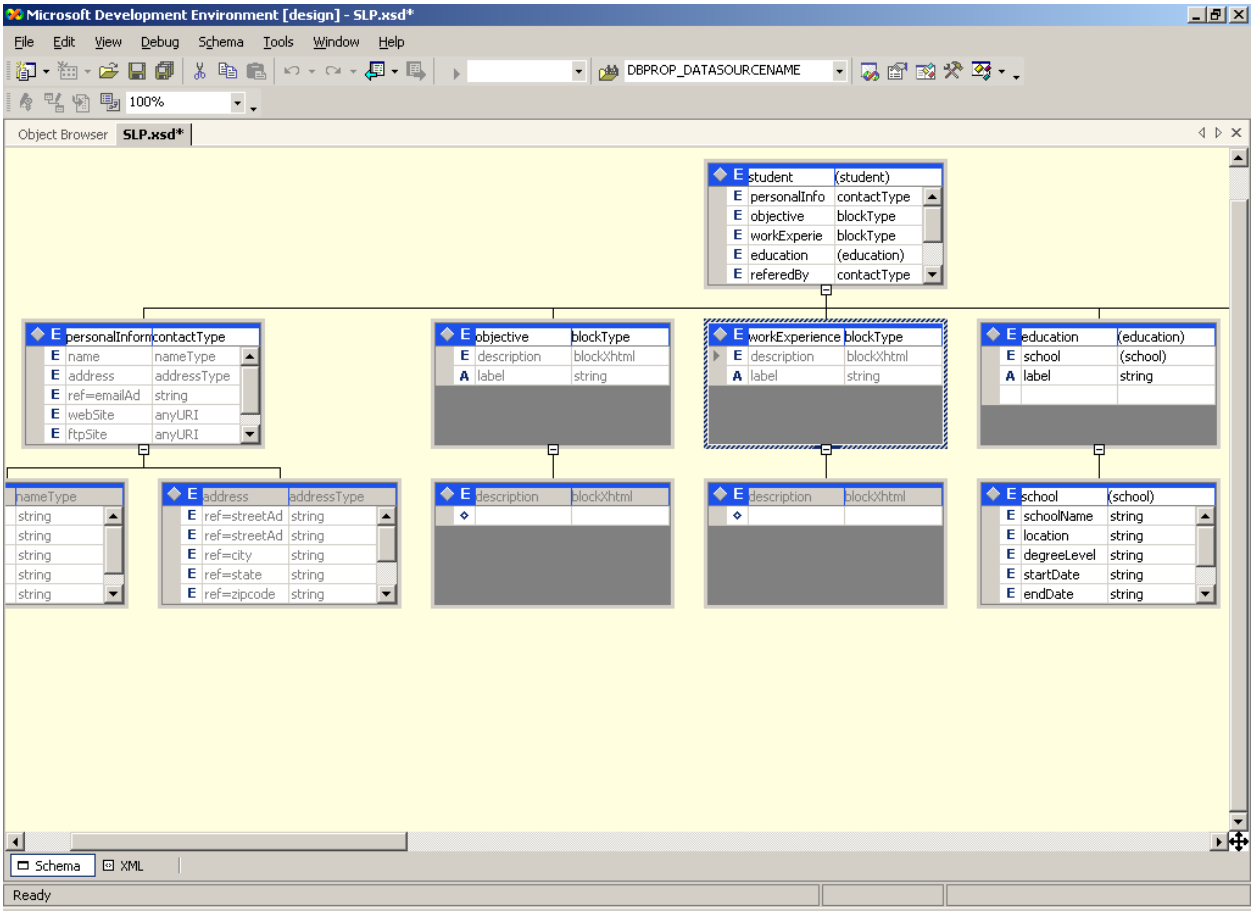

Figure 8 Schema converted with MS .NET

Addendum C: Project Schedule

Initial schedule plan

Schedule	8week2 Fall 04	8week1 Spring 05	8week2 Spring 05
Regis Courses	MSCN 690	MSCE 610 E-Security	MSC 696 A, B, C
Goals			Commencement, May, 2005
Phase one	Business Requirements Analysis		
Phase two	Specification and architecture		
Phase three	Complete first draft of MSC 696 A “Thesis Proposal” and Faculty Approval		
Phase four		Functioning graphic user interface prototype that collects documents from students and NGOs	
Phase five			Completion project documents and academic papers

Table 4 Project Schedule