An Undergraduate Web Engineering Program

Albert C. Esterline Kenneth A. Williams Edward C. Carr[i]
North Carolina A&T State University
Computer Science department
Greensboro, NC 27406
(336) 334-7245
esterlin@ncat.edu williams@ncat.edu corwith@ncat.edu

ABSTRACT

This paper presents an undergraduate web engineering curriculum under development. The need for an undergraduate web engineering program distinct from a computer science program arises from the diversity of standards to be covered, fundamental differences in the kind of engineering required, and the need to address the social dynamic of the Web. The feasibility of an undergraduate program in web engineering is supported by the coherence of the standards and the fact that the engineering and science fit naturally with the standards and technology. In general, the web standards are critical for developing the curriculum, and their coherence has been exploited in defining it. Future plans include collaborations, evolution of the curriculum, and development of a model curriculum. It is anticipated that a web engineering program will compensate for years of falling enrollment in computer science programs and the drop in the percentage of female students enrolled.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education - *Curriculum*

General Terms

Standardization

Keywords

Web Engineering, Curriculum, Undergraduate Education

1. INTRODUCTION

This paper presents and discusses a Web engineering curriculum to be initiated by the Computer Science Department of North Carolina A&T State University (NC A&T) in the 2009-2010 academic year. This program is in direct response to the importance the Web has assumed in the past decade or so. This success has been based on a largely mature if quite recent set of standards. Industry has developed software that implements these standards, and there are numerous open-source projects dedicated to developing systems based on the standards. This technology has been astonishingly fruitful in supporting applications in a vast number of areas, and the Web now underpins most of the new social and economic activity around the world. As a result, major employment opportunities exist for those able to use these languages and systems and to exploit the standards.

The standards are critical for developing a curriculum that is academically respectable, for defining a true discipline, and for relating it to existing disciplines. Web standards are parsimonious and clearly defined, and they are reminiscent of the models of computing that launched computer science. In fact, a relatively large and growing number of standards are involved. Some, such as XML, have filled needs that go beyond use of the Web. Others have become de facto standards in areas that originally had nothing to do with the Web—consider, for example, how the Web Ontology Language (OWL) has become accepted for knowledge representation. The standards bodies have ensured that the diverse standards form a coherent whole. This coherence is exploited in defining the curriculum.

Introducing a program in web engineering in a computer science department is particularly attractive now, after several years of falling enrollment in computer science programs and a precipitous drop in the percentage of female students in such programs. To counter the decline in enrollment, a web engineering program can introduce technologies that directly engage the user from the outset. Among our own undergraduate students, web and Internet courses are consistently poplar, and most of their senior projects involve web applications. In addition, the opportunities the Web provides for education are enormous, going well beyond the content management systems now typical of distance learning. Academic programs strive for a sense of community, but a program educating web engineers can develop and maintain a Web community that keeps members in touch minute-by-minute and provides access to any item or member with only a couple of clicks. Learning in large part can be problem driven, and each course can have at least one substantial project that uses the Web or web technologies. Finally, a web engineering program should have more appeal for female students than traditional computer science programs since

it addresses social phenomena substantially and de-emphasizes "under-the-hood" wizardry.

The remainder of this paper is organized as follows. Section 2 explains why a web engineering program is distinct from a computer science program. Section 3 discusses the feasibility of an undergraduate program in web engineering. The curriculum and the proposed course are described in section 4. Future plans for collaboration and expansion are given in section 4 while section 5 provides the conclusion.

2. NEED FOR A DISTINCT PROGRAM

The need for a web engineering program distinct from a computer science program is most readily demonstrated by considering how the standards stack up. In the curriculum laid out in section 4, there is a prerequisite chain of seven semester courses. Some of the prerequisite relations, such as the course on web services requiring the course that thoroughly addresses XML Schema, follow directly from the relations among the standards. Other prerequisite relations have pedagogical motivation that exploits relations among standards, such as requiring knowledge of HTML before proceeding to XML. In any case, fitting a prerequisite chain of seven courses into a computer science program would simply change the nature of the program. Another reason for a distinct program is that, as has often been noted, engineering web applications is different in kind from engineering more traditional software applications. While a separate course on engineering web applications could make students aware of the differences, a better policy is to investigate engineering issues in the same courses in which web standards and programming are introduced so that engineering skills may evolve with practice and build in a way reflecting the relationships among the standards.

Possibly the most fundamental reason to have a separate web engineering program, however, is that understanding the Web as an entity requires understanding more than technological issues; it requires understanding the social dynamic inherent in it. To successfully engineering web applications and systems in a principled way, then, requires an understanding of these social aspects. This view is being pursued under the rubric of 'web science" [1], notably in a joint effort of MIT and the University of Southampton. The social nature of the Web is the focus of a nexus of concepts and attitudes identified as Web 2.0, which emphasizes collaboration among users. As with web engineering, a separate course could be dedicated to web science, but a better approach is to work the science in with the introduction of standards and technologies. After all, the success of these standards and technologies rests in the social dynamic that they have launched.

It is worth noting that the curriculum presented in Section 4 includes a junior-year course Social and Economic Aspects of the Web, which is the sole course in the curriculum dedicated to social aspects. This course will foster awareness of issues and present techniques for analyzing situations. Senior-year classes then will provide extensive opportunities to apply the techniques and exercise the awareness. In fact, the very first course in the curriculum will introduce the use of tools in the vein of Web 2.0, which will be used throughout the program to maintain a community.

Finally, from a more general perspective, very strong motivation for the program arises from the fact that the Web as a social and economic phenomenon of enormous magnitude simply demands to be addressed at all levels. The L.I.K.E.S. (Living In the KnowEdge Society) Community Building Project, in which NC A&T is a partner, is funded by the National Science Foundation through its C-PATH initiative to improve awareness and interest in computing. This project aims to transform computing education for the twenty-first century, and the community is identifying key concepts from computing and other disciplines that will be pivotal for disseminating knowledge [5]. The central role of the Web across virtually all disciplines and, indeed, social and economic endeavors that emerged from these investigations encouraged us to develop the program presented here.

3. FEASIBILITY

While courses in web engineering topics are offered by most computer science departments, we are unaware of any complete undergraduate web engineering major being offered at any university. The University of Western Sydney has offered a successful web engineering track in its masters program since 1999 [2], and Yogesh Deshpande from Western Sydney has proposed a framework to design curricula for web engineering that can evolve in tandem with the evolution of the Web [3]. In addition, E. James Whitehead from the University of California, Santa Cruz identified the relevant knowledge areas and laid out a proposal for a graduate program in web engineering [9].

It might be thought that web engineering is too involved and sophisticated to be the subject of an undergraduate program. Some of the largest software projects these days are centered on the Web, the social aspects are still largely research topics, and,

indeed, existing programs in Web engineering are at the graduate level. The positive response to these doubts is implicit in the above discussion. The standards themselves are elegant and coherent. Gaining familiarity with them does not require enterprise-level projects, and the commonly used server-side scripting languages all support these standards. Because of the nature of the Web, learning to use the standards and technologies necessarily involves principled approaches normally classified as engineering. Finally, going a step further, understanding why some things work and some do not involves considering social aspects in a concrete way. Insight here can be gained by introducing techniques normally taught at the undergraduate level (albeit not in computer science), such as game theory, and techniques that leverage background knowledge, such as social network analysis, based on graph theory. Generally, then, it will be critical to the success of the undergraduate program that the engineering and science fit naturally with the standards and technology.

Note that some of the current and envisioned elaboration of the Web is in how standards are applied rather than the standards themselves. Thus, Web 2.0 uses the basic web standards in a consciously social way, and the foundation of Web 3.0, the Intelligent Web, are the standards defining the Resource Description Framework (RDF) and OWL for the Semantic Web. These standards are quite mature, and, again, the main goal for a student is to appreciate their potentials.

4. CURRICULUM

The educational objectives of the web engineering program are that graduates will:

- be able to develop sophisticated web applications, including machine-to-machine applications, that require extensive engineering based upon sound methodologies and standards,
- be able to maintain a sophisticated, secure web site,
- be able to lead a web development team,
- understand the social and economic potential of the Web, its future, and problems it faces,
- be able to analyze social and economic Web activity in terms of individual players as well as in aggregate terms,
- keep up to date with new web standards and technologies and participate in their future development, and
- be prepared to fulfill their social, legal, ethical and professional obligations as web-engineering professionals.

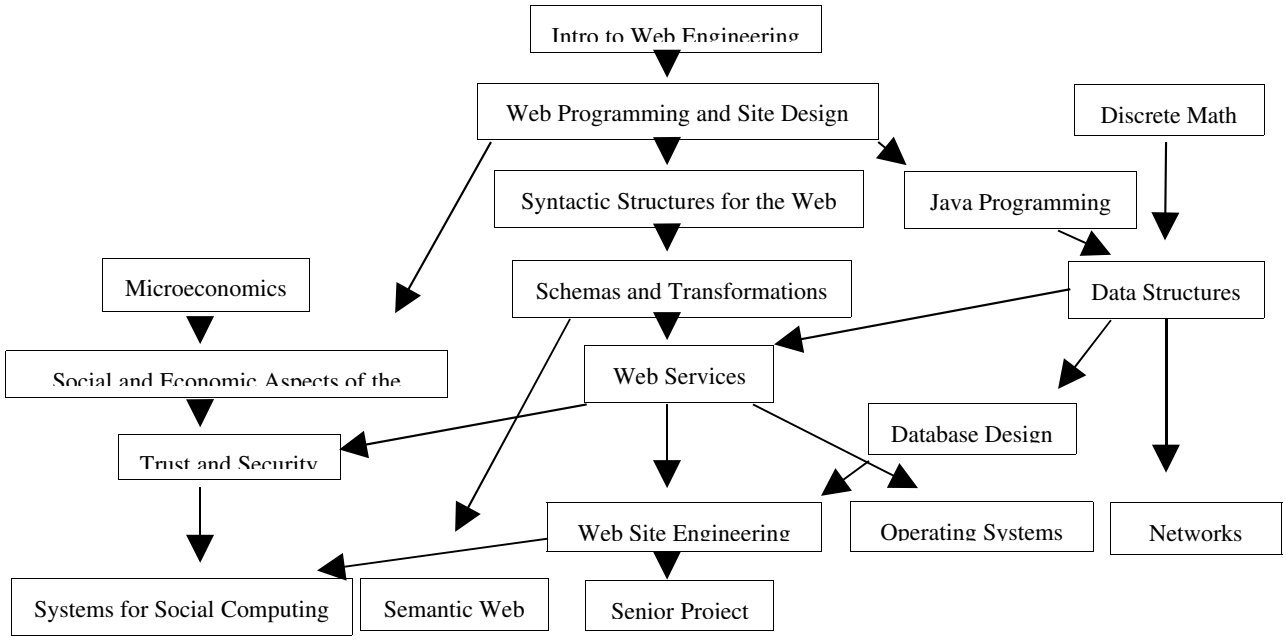

Figure 1 – Prerequisite chart for required web engineering courses. Existing courses in computer science, math and economics are shown in dashed boxes.

The curriculum is designed to meet the accreditation requirements of ABET's Engineering Accreditation Commission under the Software Engineering criteria [4]. Students majoring in web engineering are required to take a total of 124 semester hours, of which 37 hours are required web engineering courses. All web engineering courses are three semester hours except the Semantic Web course, which is two hours, and the two freshmen courses, which are four hours each due to the attached lab periods. The required web engineering courses are listed below. The prerequisite graph for required web engineering courses is given in figure 1.

Introduction to Web Engineering

This course introduces basic web development using HTML and client-side and server-side scripting. Students also learn how to incorporate security features into web sites as well as how to access and manage online databases. Students gain hands-on experience using web-based technologies and they come to understand the role of the Web in disseminating knowledge, community formation, training, collaboration, and other social activities.

Web Programming and Site Design

This course gives a systematic presentation of the XHTM standard for web pages, the HTTP protocol, client-side and server-side scripting, style sheets for presentation, web access to databases, and AJAX.

Syntactic Structures for the Web

This course presents the Extensible Markup Language (XML), used for sharing structured data and documents across the Web. Theoretical foundations in formal language theory provide principled ways to understand and to design XML documents and XML-derived markup languages.

Schemas and Transformations

This course teaches how to define an XML-based language by producing a schema that defines the structure of valid documents in that language. It also teaches standards for transforming the structure of a document.

Object-Oriented Programming and Design

This course teaches the language of instruction of the Computer Science program so that Web-Engineering students may take courses from that program. Language coverage in this course is comparable to that of a two-semester sequence for Java as a first language.

Social and Economic Aspects of the Web

This course uses methods from the social sciences and economics to analyze the Web as a social and economic phenomenon, providing an understanding of how the Web enables online communities, is a medium for commerce, and makes possible a host of socio-economic activities on an unprecedented scale.

Trust and Security

This course analyzes trust, including the competence, predictability, benevolence, and integrity of another party in Web-enabled relations. More specifically, this course addresses Web-related security issues and technologies.

Web Services

This course addresses interoperable machine-to-machine interaction over the Web. The focus is on various standards for exchanging XML-based messages, for describing services, and for listing and discovering services.

Semantic Web

This course presents the extensions to the Web in which the semantics of Web content is defined to facilitate its use by people and machines. A range of specifications is addressed for encoding ontologies, that is, for formally describing the concepts and their relations within a domain.

Web Site Engineering

The design and implementation of coherent web sites that provide various types of data and services in various forms are covered in this course. Of particular interest are standards, practices, and software for aggregating and mashing up content.

Systems for Social Computing

This course addresses the design, implementation, and use of systems that support the collection, processing, and dissemination of information that is significant because it is produced and consumed by people linked by a social network.

Senior Project

This course provides the opportunity to design and implement a Web project of significant size from start to finish.

In addition to the required web engineering courses, students are required to take

- Nine hours of technical electives, which are generally either web engineering or computer science courses
- Computer science classes in data structures, operating systems, data base design and networks, totaling 12 hours
- Discrete Mathematics I & II, Calculus I & II, Probability and Statistics and an advanced mathematics course, for a total of 20 hours of mathematics
- Twelve hours of natural sciences
- Three hours of microeconomics
- General education courses totaling 28 hours

In outline, then, the program leads the student in the first three years from technologies that directly engage the user to web services, which assume no direct human involvement but have become the dominant technology. Freshman-year courses, besides engaging the students with web technologies, teach them well-engineered client-server computing that involves visually-rendered HTML documents, thereby teaching them two scripting languages (one for the client side, one for the server side). The sophomore year takes the students through the first semester of the sophomore year of the Computer Science program. It also covers web standards and technologies that are more fundamental, but less visually oriented, than web programming that serves up HTML documents. The junior year presents web services, which exploit the standards covered in the previous year, and it covers social aspects of the Web, required for senior-year topics. Finally, the senior year covers senior-year computer science topics intimately related to the Web and some advanced standards for realizing the vision of the Web. It also involves large web applications, requiring sound engineering, mastery of the various layers of web technology, and an understanding of the social role of the Web.

5. FUTURE PLANS

In the future, we intend to collaborate with industry, organizations, and other academic institutions to enhance the curriculum and to champion undergraduate education in web engineering. We also intend to introduce more web science into the curriculum, to facilitate student transfer into the program, and to promulgate the curriculum as a model for undergraduate web engineering.

The Web's growth has been driven in large part by industry's efforts to exploit the unprecedented commercial opportunities it provides. To keep in touch with the industrial roots, we intend to establish an Industrial Advisory Board for the program modeled on the successful similar board for our Computer Science program. The open nature of the Web, however, has given key roles to certain organizations, most obviously standards organizations, but also, and derivatively, open-source software foundations and societies that explicitly promote web engineering. The influence of such organizations argues for permanent liaisons with them. In addition, we seek interaction with other academic institutions interested in establishing similar programs or interested more generally in enabling the Web's potentials (such as our partners in the LIKES community mentioned above).

The curriculum presented in section 4 has just one course dedicated to topics directly addressing web science, although we noted that such topics will be introduced as relevant technologies are taught. An undergraduate curriculum is constrained by accreditation requirements and the need to provide a sound and thorough foundation in the standards and technologies. These constraints unfortunately leave little room for core courses lying off the technology backbone. What is feasible is to develop electives dedicated to topics from web science. A consolation

here is that material that matures in delivering the electives can be selectively transferred into core courses.

The web engineering program would be particularly open to transfer students. A precedent is being established with The North Carolina Community College System (NCCCS), which introduced the Web Technologies curriculum program in the spring of 2006 to supersede its Internet Technologies curriculum program (approved in 1999) [6]. It has been recognized as a subject for which certificates, diplomas, and Associate in Applied Science (AAS) degrees may be awarded [7], and 43 of the 58 constituent offer this degree program [8]. Negotiation is underway to establish an articulation agreement so that NCCCS students may transfer their credits when they enroll in the web engineering program at NC A&T. In fact, more intimate connections than pipelining students are planned. In particular, part of the program will involve an online community not only to facilitate delivery of the program but also to immerse students in the Web. This community will be extended to NCCCS students. Indeed, students from nearly any academic institution could be enlisted into the community.

Finally, we intend to pursue the development of a curriculum standard for Web engineering. It would be appropriate for web engineering to eventually become part of the computing disciplines defined by the ACM/IEEE-CS Computing Curricula along with computer science, information systems, information technology, computer engineering, and software engineering.

6. CONCLUSION

This paper has discussed the web engineering curriculum being initiated by the Computer Science Department of NC A&T; the curriculum was outlined in section 4. It was argued that the web standards are critical for developing the curriculum, and their coherence has been exploited in defining the curriculum. The need for a web engineering program distinct from a computer science program was addressed. This need, it is claimed, arises from the size and diversity of the standards to be covered, the different kind of engineering required, and the fact that web engineering must address the social dynamic of the Web. The feasibility of an undergraduate program in web engineering is supported by the elegance and coherence of the standards and the fact that the engineering and science fit naturally with the standards and technology. It is anticipated that a web engineering program will compensate for years of falling enrollment in computer science programs and the drop in the percentage of female students enrolled. Finally, in section 5, we outlined our plans for the future. These include collaborations to enhance the program, introducing more material related to web science, facilitating student transfer into the program, and developing an ACM/IEEE model curriculum. We are confident that program graduates will have the background needed for mastering the scientific, technical and social challenges of the Web.

7. REFERENCES

[1] Berners-Lee, Tim, Hall, Wendy, Hendler, James, Shadbolt, Nigel, Weitzner, Daniel J. 2002. Creating a science of the Web. Science 313, (Aug. 2006), 769-771.

[2] Deshpande, Yogesh, Murugensan, San, Ginige, Athula, Hansen, Steve, Schwabe, Daniel, Gaedke, Martin, and White, Bebo. 2002. Web Engineering. Journal of Web Engineering. 1, 1. 3-17.

[3] Deshpande, Y. Web engineering curriculum: a case study of an evolving framework. ICWE 2004. 526-530.

[4] Engineering Accreditation Commission. 2007. Criteria for Accrediting Computing Programs. ABET, Inc.: Baltimore, MD. http://www.abet.org/Linked Documents-UPDATE/Criteria and PP/C001 08-09 CAC Criteria 11-8-07.pdf

[5] Living in the KnowlEdge Society Community Building Project. Introduction. (Aug. 2008). http://www.likes.org.vt.edu/

[6] North Carolina Community College System. 2007. Combined Course Library: Web Technologies. http://www.ncccs.cc.nc.us/Programs/docs/Common-Course-Library/W_prefix/WEB_12Sep2007.pdf

[7] North Carolina Community College System. 2008. Curriculum Standard: Web Technologies. http://www.ncccs.cc.nc.us/Programs/docs/Curric_Standards/25/A25290 Web Tech SP06.pdf

[8] North Carolina Community College System. 2008. Education Catalog (March 14, 2008). http://www.ncccs.cc.nc.us/Programs/docs/Education-Catalog/Education_Catalog_03-14-2008.doc

[9] Whitehead, E. James, Jr. 2001. A proposed curriculum for a masters in web engineering. In Proceedings of the Fourth Workshop on Web Engineering, held with WWW10 (Hong Kong, May 1, 2001).

[i] Supported in part by NSF grant CCF 0722289 Living in the Knowledge Society (LIKES)