A Draft Web Engineering Curriculum Pattern

Antonio Navarro

Dpto. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid
28040, Madrid, Spain
anavarro@fdi.ucm.es

Abstract. The implementation of higher education studies is not a simple matter. However the presence of specialized curriculum recommendations can make implementation easier. Web engineering is an important discipline that is well worth the implementation of higher education studies, but, at present, no specialized curriculum recommendations on this subject have been developed. This paper gives voice to some thoughts about web engineering and its relationship with software engineering and information systems. In addition, a draft curriculum recommendation for web engineering is proposed.

Keywords: curriculum recommendation, software engineering, information systems

1 Introduction

The implementation of higher education studies is far from being a simple matter. In recent years, the development of the *European Space for Higher Education* [1] has promoted the renewal of higher education curricula in several European countries. For example, in Spain, the whole structure of university studies has changed. Thus, degrees with a duration of three and five years have been replaced by degrees lasting four years plus Master's studies.

Indeed, our own university has changed its curricula and now offers three different degrees in computer science. During the development of these degrees, the presence of computer science curricula authored by ACM/AIS/AITP/IEEE-CS [2] facilitated implementation to a significant extent.

Web engineering is an important discipline that is attracting the interest of students, academics and industry [3, 4, 5, 6, 7, 8, 9, 10, 11]. Therefore, the implementation of higher education studies in web engineering is a reasonable effort that ought to be carried out by educational institutions. However, at present, no specialist curriculum recommendations on web engineering have been put forward by any international organization. Only some isolated efforts have been made by individuals [12, 13, 14]

One major obstacle in the development of these specialist curriculum recommendations on web engineering is agreement about what web engineering is. Some authors hold that web engineering is a new emerging discipline in its own right, rather than one that is subsumed under software engineering [5, 15]. However, this finding does not in itself define the nature of web engineering. Taking into account the different disciplines that make up web engineering [4, 5, 9] a set of constituent disciplines has been proposed [16].

This paper reviews these constituent disciplines and their relationships with available curriculum recommendations and expresses some thoughts about the development of curriculum recommendations on web engineering. For the sake of conciseness, only curriculum recommendations from ACM/AIS/AITP/IEEE-CS are reviewed.

Thus, in Section 2 these constituent disciplines are reviewed. In Section 3 some ACM/AIS/AITP/IEEE-CS curriculum recommendations are reviewed. In Section 4 a draft curriculum pattern for the web engineering discipline is defined. Finally, our conclusions and future work are outlined.

2 Constituent Disciplines of the Web Engineering Discipline

Several authors have proposed that web engineering is a discipline made up of different heterogeneous disciplines [4, 5, 9]. The study referred to in [16] takes these proposals into account, as well as some topics included in different journals and conferences focusing on web engineering, and proposes a set of *constituent disciplines* of web engineering disciplines. These disciplines are:

- Accessibility.
- Agents.
- Document and text processing.
- e-commerce.
- e-learning.
- Hypermedia and hypertext.
- Human-computer interaction.
- Information systems.
- Programming languages.
- Semantic web.
- Software engineering.
- Web engineering fundamentals.
- Security.
- Others.

However, these disciplines do not have the same relevance in the web engineering discipline. In order to obtain an approximate idea of the influence of every constituent discipline more than seven hundred papers on web engineering were classified in [16]. Two classifications were made: (i) considering only one main indexing discipline; and (ii) considering up to three indexing disciplines. Table 1 displays the results.

Table 1. Number of papers per constituent discipline of web engineering [16]

Discipline	**Number of papers (percentage) considering one main indexing discipline**	**Number of papers (percentage*) considering up to three indexing disciplines**
Software Engineering	279 (38.9%)	329 (45.8%)
Information Systems	98 (13.6%)	211 (29.4%)
Semantic web	68 (9.5%)	82 (11.4%)
Human-Computer Interaction (HCI)	31 (4.3%)	52 (7.2%)
Document and text processing	29 (4.0%)	64 (8.9%)
e-learning	29 (4.0%)	31 (4.3%)
Agents	22 (3.1%)	28 (3.9%)
Hypermedia and hypertext	20 (2.8%)	33 (4.6%)
e-commerce	16 (2.2%)	21 (2.9%)
Security	16 (2.2%)	24 (3.3%)
Accessibility	13 (1.8%)	22 (3.1%)
Web engineering fundamentals	9 (1.3%)	9 (1.3%)
Programming languages	6 (0.8%)	10 (1.4%)
Others	82 (11.4%)	82 (11.4%)
TOTAL	718 (100%)	n/a

*percentage of 718 papers

The previous analysis does not attempt to conduct an exhaustive classification and indexing of web engineering literature. The analysis does not aim to precisely rank constituent disciplines of web engineering by order of importance either. Its principal aim is to obtain an approximate view of the importance of every constituent discipline in web engineering. Thus, the analysis clearly indicates that software engineering is a very important part of the web engineering discipline. In addition, it indicates that information systems are another important discipline in web engineering. But of course, these are not the only constituent disciplines.

Therefore, it is reasonable to derive from [16] that software engineering and information systems are two of the most important disciplines in web engineering. Fortunately, ACM/AIS/AITP/IEEE-CS have provided curriculum recommendations for these two disciplines. The next section provides a brief review.

3 Curriculum Recommendations

ACM/AIS/IEEE-CS Computing Curricula 2005 (CC2005) provides undergraduate curriculum guidelines for five defined sub-disciplines of computing: Computer Science, Computer Engineering, Information Systems, Information Technology, and Software Engineering [2]. In addition, specific curriculum recommendations are provided for every sub-discipline of computing. This section briefly reviews the curriculum recommendations for software engineering and information systems.

3.1 ACM/IEEE-CS Software Engineering 2004

ACM/IEEE-CS Software Engineering 2004 (SE2004) provides curriculum guidelines for undergraduate degree programs in software engineering [17]. SE2004 defines SEEK (*Software Engineering Education Knowledge),* the body of knowledge that is appropriate for an undergraduate program in software engineering. SEEK is divided into knowledge areas, units and topics. The knowledge areas identified by SE2004 are: computing essentials, mathematical and engineering fundamentals, professional practice, software modelling and analysis, software design, software verification and validation, software evolution, software process, software quality, software management. In addition, SE2004 identifies different curriculum patterns, described in terms of courses, which ultimately cover topics.

Table 2 depicts a curriculum pattern for a software engineering degree taught in a computer science department.

Table 2. Curriculum pattern N2S-1c - in a computer science department [17]

Year 1		Year 2		Year 3		Year 4	
Sem 1A	Sem 1B	Sem 2A	Sem 2B	Sem 3A	Sem 3B	Sem 4A	Sem 4B
CS101	CS102	CS103	CS220	CS226	CS270T	SE400	SE400
Calc 1	*Calc 2*	CS106	SE A	MA271	SE D	SE F	*Tech elect*
NT181	CS105	SE201	SE212	SE C	SE E	*Tech elect*	*Tech elect*
Physics	*Any sci*	NT272	*Linear Al*	NT291	*Tech elect*	*Tech elect*	*Tech elect*
Gen ed	*Gen ed*	--	*Gen ed*	*Gen ed*	*Gen ed*	*Gen ed*	*Gen ed*

CS101:Programming fundamentals; CS102: Object-oriented paradigm; CS103: Data structures and algorithms; CS105: Discrete structures I; CS106: Discrete structures II; CS220: Computer architecture; CS226: Operating systems and networking; CS270T: Databases; *Calc n*: Calculus n; SE 201: Introduction to SE; SE212: SE Approach to HCI; SE A, C, D, E, F: additional SE courses; SE400: SE capstone project; MA271: Statistics and empirical methods for computing; NT181: Group dynamics and communication; NT272: Engineering economics; NT291: Professional SE practice; *Tech elect*: technical elective course; *Gen ed*: general non-technical course.

3.2 ACM/AIS/AITP Information Systems 2002

ACM/AIS/AITP Information Systems 2002 (IS2004) provides curriculum guidelines for undergraduate degree programs in information systems [18]. The architecture of the information systems curriculum at the highest level consists of five curriculum presentation areas: information systems fundamentals, information systems theory and practice, information technology, information systems development, and information systems deployment and management processes. The five presentation areas consist of ten courses and one prerequisite course.

Fig. 1 depicts these ten courses and their dependences.

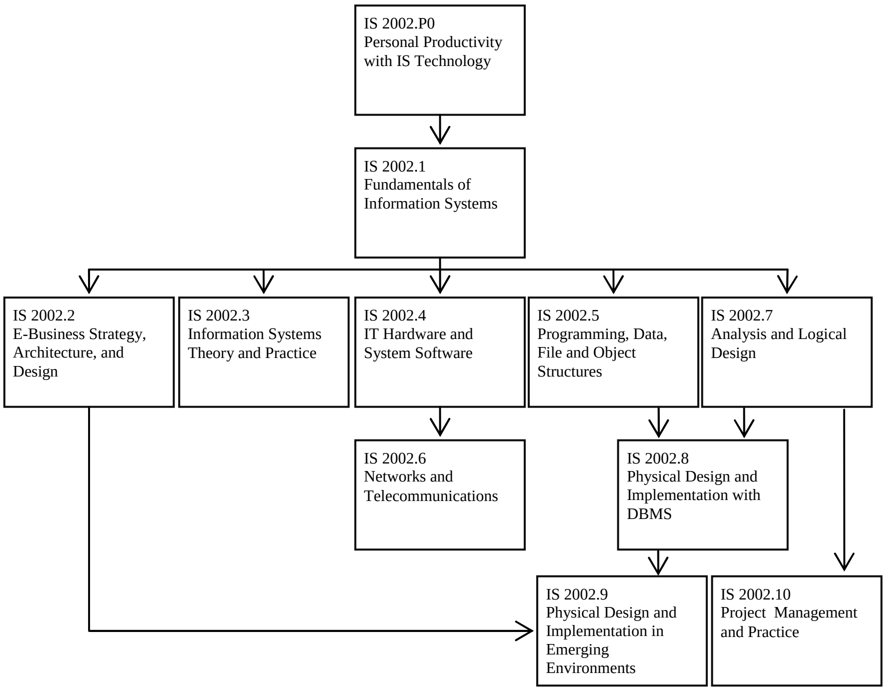
IS 2002.P0
Personal Productivity with IS Technology
IS 2002.1
Fundamentals of Information Systems
IS 2002.2
E-Business Strategy, Architecture, and Design
IS 2002.3
Information Systems Theory and Practice
IS 2002.4
IT Hardware and System Software
IS 2002.5
Programming, Data, File and Object Structures
IS 2002.7
Analysis and Logical Design
IS 2002.6
Networks and Telecommunications
IS 2002.8
Physical Design and Implementation with DBMS
IS 2002.9
Physical Design and Implementation in Emerging Environments
IS 2002.10
Project Management and Practice

Fig. 1. Courses in IS2002 and their dependences [18]

4 A Draft Web Engineering Curriculum Pattern

4.1 Curriculum Pattern

The development of a curriculum pattern is not an easy task. The patterns provided in the curriculum recommendations put forward by ACM/AIS/AITP/IEEE-CS are large documents made by teams of experts in every area. Therefore, the curriculum pattern provided in this paper is not comparable to the curriculum patterns contained in ACM/AIS/AITP/IEEE-CS recommendations. However, taking into account the work done in [16], and the excellent work carried out in ACM/AIS/AITP/IEEE-CS recommendations, it is reasonable to propose a draft curriculum pattern that merges the curricula provided by SE2004 and IS2002. Table 3 displays such a curriculum pattern.

Table 3. A Draft Web Engineering Curriculum Pattern

Year 1		Year 2		Year 3		Year 4	
Sem 1A	Sem 1B	Sem 2A	Sem 2B	Sem 3A	Sem 3B	Sem 4A	Sem 4B
CS101	CS102	CS103	CS220	CS226	SE D	SE400/IS2002.10	SE400/IS2002.10
Calc 1	*Calc 2*	CS106	SE A	MA271	SE E	SE F	IS2002.8
NT181/NT272	CS105	SE201	SE212	SE C	IS2002.7	IS2002.6	IS2002.9
Physics	*Linear Al*	IS2002.1	IS2002.2	IS2002.3	WE202	WE3xx	WE3xx
Gen ed	IS2002.P0	WE101	NT291	WE201	*Gen ed*	WE3xx	*Gen ed*

Regarding the software engineering curriculum pattern described in Table 2 and the information systems courses described in Fig. 1:

- NT272 Engineering economics has been moved from Sem 2A to Sem 1A and has been merged with NT181 Group dynamics and communication.
- NT291 Professional SE practice has been moved from Sem 3A to Sem 2B.
- Linear Algebra has been moved from Sem 2B to 1B.
- CS270T Databases has disappeared. The inclusion of courses belonging to IS2002 makes the presence of course CS270 on databases unnecessary.
- IS2002.4 IT hardware and system software has disappeared. The presence of computer science courses (cs22x) makes course IS2002.4 on computer architecture unnecessary.
- IS2002.5 Programming, data, file and object structures has disappeared. The presence of computer science courses (CS10x) makes course IS2002.5 about programming and data structures unnecessary.
- SE400 SE capstone project and IS2002.10 Project management and practice have been merged. The capstone project on software engineering can be merged with the capstone project on information systems. Of course, the project should be a web engineering project.

In addition, specific courses on web engineering have been added. These courses have been selected taking into account the skills needed for the development of web applications and the constituent disciplines of web engineering identified in [16]:

- WE101 Web engineering fundamentals. This introductory course presents the web and introduces fundamental concepts such as: HTML, HTTP protocol, web servers, etc. Shklar's book [19] can be a valuable reference for this course.
- WE201 Web programming. This course aims to teach how to build web applications using an advanced programming language. For example, if J2EE is chosen, [20, 21] can be valuable references. WE101 is a prerequisite for this course.
- WE202 Components and service oriented architectures. This is an advanced course about distributed components and service oriented architectures. Again, if J2EE is chosen, [22, 23] can be valuable references. WE201 is a prerequisite for this course.
- Courses WE3xx: These are courses on specific web engineering topics [16]. Because the capstone project is under development in the last year we

think that the teaching load should be lightened; these, therefore, could be introductory courses. Master's and doctoral programs could provide deeper descriptions of these courses. In particular, we propose ten courses with (at least) a representative reference:

- WE301: Accessibility [24].
- WE302: Agents [25].
- WE303: Document and text processing [26].
- WE304: e-commerce [27]
- WE305: e-learning [28].
- WE306: HCI for the web [29, 30].
- WE307: Hypermedia and hypertext [31].
- WE308: Security [32].
- WE309: Semantic web [33].
- WE310: Web 2.0 [34].

Software engineering courses should be reviewed according to [16] to include specific web engineering issues. A similar effort should be made with information systems courses. Thus, these courses should pay attention to specific web engineering issues such as: web project management; web design, site design and development; and legal and ethical issues (copyright). In addition, the duration of every course could be changed in order to fulfil the scheduling requirements of every educational institution.

The web engineering curriculum pattern depicted in Table 3 may be a bit dense. Removing advanced courses in software engineering and information systems could be a simple way to lighten it. General non-technical courses could be taught instead.

SE2004 also includes significant additional sections: Software Engineering as an Engineering Discipline, Professional Practice, Guiding Principles (for the development of SE2004) and Student Outcomes. We agree with the underlying philosophical message of these sections. Therefore, they could be adapted and included in a curriculum recommendation for web engineering.

IS2002 also includes significant additional sections: Principles Guiding the Curriculum Design, Guiding Assumptions about the Information Systems Profession, and Exit Characteristics of Information Systems graduates. As in the case of SE2004, we agree with the underlying philosophical message of these sections. Therefore, they could be adapted and included in a curriculum recommendation for web engineering.

4.2 Related Web Engineering Curriculum Recommendations

Regarding the available web engineering curriculum recommendations, it is important to observe that, as the curriculum proposed in this paper, none is made by a major organization.

The curriculum proposed in this paper is an undergraduate degree program, while the curriculum proposed in [12] is a Master's program. Both programs lay emphasis on software engineering and information systems. [12] also identifies Networking as an important element. This element is covered in the curriculum proposed in this paper by courses CS226 Operating systems and networking and IS2002.6 Networks and telecommunications.

A framework for a web engineering curriculum is defined in [13]. Therefore, the work carried out in [13] is more abstract than that presented in this paper. However, the courses proposed in this paper fit quite well into the framework proposed in [13].

Finally, in [14] an undergraduate web engineering program is also proposed. The courses proposed in this paper are in tune with the courses proposed in [14]. However, eighteen courses are proposed in [14], while more than forty courses are proposed in this paper. Therefore, our approach is more detailed.

5 Conclusions and future work

The development of a curriculum recommendation is not an easy task. According to the constituent disciplines of web engineering, this paper has proposed a draft curriculum pattern for web engineering.

This curriculum pattern could be reviewed and lightened to be made more attractive for both students and teachers. In addition, software engineering and web engineering courses must be adapted to the specific nature of the web engineering discipline.

However a question arises on analyzing the proposed curriculum pattern: is a web engineer a programmer with a solid foundation in software engineering and information systems that is able to build web applications and is aware of technologies used in web development? If the answer is yes, the proposed curriculum could serve as a basis for the development of web engineering curriculum recommendations. Otherwise, a concise definition of what a web engineer is should be formulated.

A final conclusion is that, in our opinion, the course structure presented in SE2004 is more useful and detailed than the one presented in IS2002. Therefore, if a web engineering curriculum recommendation is made, the structure defined in SE2004 would be preferable to the one defined in IS2002.

Finally, we think that future work should include the development of web engineering undergraduate curriculum recommendations by a major worldwide organization. In addition, the development of Master’s and doctoral programs should be analysed.

Acknowledgments. *El Ministerio de Ciencia e Innovación* (TIN2009-14317-C03-01/TSI), *La Comunidad Autónoma de Madrid* (S2009/TIC-1650) and *La Universidad Complutense de Madrid* (Group 921340) have supported this work.

References

1. The Bologna Declaration on the European space for Higher Education: An Explanation, http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf

2. ACM/AIS/IEEE-CS Computing Curricula 2005. The Overview Report, http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf (2005).
3. Deshpande, Y., And Hansen, S.: Web Engineering: Creating a Discipline among Disciplines. IEEE MultiMedia 8, 82--87 (2001).
4. Deshpande, Y. Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., White, B.: Web Engineering. Journal of Web Engineering 1, 3--17 (2002).
5. Ginige, A., and Murugesan, S.: Guest Editors' Introduction: The Essence of Web Engineering-Managing the Diversity and Complexity of Web Application Development. IEEE MultiMedia 8, 22--25 (2001)..
6. Kappel, G., Pröll, B., Reich, S., Retschitzegger W. (Eds.): Web Engineering: The Discipline of Systematic Development. John Wiley & Sons Inc., Chichester (2006).
7. Mendes, E., Mosley, N. (Eds.): Web engineering. Springer-Verlag, Berlin (2005).
8. Murugesan, S., Deshpande, Y. (Eds.): Web Engineering: Managing Diversity and Complexity of Web Application Development. LNCS 2016, Springer-Verlag, Berlin (2001).
9. Murugesan, S., Deshpande, Y., Hansen, S., Ginige, A.: Web Engineering: A New Discipline for Development of Web-Based Systems. In Web Engineering: Managing Diversity and Complexity of Web Application Development, S. Murugesan, and Y. Deshpande, Eds. LNCS 2016, Springer-Verlag, Berlin, (2001).
10. Rossi, G., Pastor, O., Schwabe, D. Olsina, L. (Eds.): Web Engineering: Modelling and Implementing Web Applications. Springer (2007).
11. Suh, W. (Ed.).: Web Engineering Principles and Techniques. Idea Group Publishing, Hershey (2005).
12. Whitehead, E.J. Jr.: A proposed curriculum for a masters in web engineering. Journal of Web Engineering 1(1), 18--22 (2002).
13. Deshpande, Y.: Web engineering Curriculum: A case Study of an Evolving Framework. ICWE 2004, LNCS 3140, 526--530 (2004).
14. Esterline, A.C., Williams, K.A., Carr, E.C.: An Undergraduate Web Engineering Program, http://redux.comp.ncat.edu/carr/web_engineering/SIGCSE_Web.pdf
15. Pressman, R.S.: Software Engineering: A Practitioner's Approach. 6th edition. McGraw-Hill, New York, (2004).
16. Navarro, A.: A SWEBOK-based Viewpoint of the Web Engineering Discipline. Journal of Universal Computer Science, 15(17) 3169--3200 (2009).
17. ACM/IEEE-CS Software Engineering 2004, http://sites.computer.org/ccse/SE2004Volume.pdf (2004).
18. ACM/AIS/AITP IS 2002 http://www.acm.org/education/education/curric_vols/is2002.pdf (2002).
19. Shklar, L., Rosen, R.: Web Application Architecture: Principles, Protocols and Practices. John Wiley & Sons Inc., Chichester, (2003).
20. Hall, M., Brown, C:. Cores Servlets and JavaServer Pages: Core Technologies, Vol. 1 (2nd edition). Prentice Hall (2003).
21. Hall, M., Brown, C., Chaikin, Y.: Core Servlets and JavaServer Pages: Advanced Technologies, Vol. 2 (2nd edition). Prentice Hall (2007).
22. Rubinger, A.L., Burke, B., Monson-Haefel, R.: Enterprise JavaBeans 3.1. O'Reilly (2010).
23. Kumar, B.,V. Narayan, P., Ng, T.: Implementing SOA Using Java EE. Addison-Wesley Professional (2010).
24. Thatcher, J. et al.: Web Accessibility: Web Standards and Regulatory Compliance. friends of ED (2006).
25. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley (2009).
26. Holzner, S.: XML: A Beginner's Guide: Go Beyond the Basics with Ajax, XHTML, XPath 2.0, XSLT 2.0 and XQuery. McGraw-Hill Osborne Media (2008).
27. Laudon, K., Guercio Traver, C. E-commerce 2010. Prentice Hall (2009).

28. Carliner, S., Shank, P. The E-Learning Handbook: A Comprehensive Guide to Online Learning. Pfeiffer (2008).
29. McCraken D., Wolfe R.J., Spool, J.M.: User-Centered Website Development: A Human-Computer Interaction Approach. Prentice Hall PTR (2003).
30. Nielsen J., Pernice, K.: Eyetracking Web Usability. New Riders Press (2009).
31. Lowe, D., Hall, W.: Hypermedia and the Web: An Engineering Approach. Wiley (1999).
32. Daswani, N., Christoph, K., Kesavan, A.: Foundations of Security: What Every Programmer Needs to Know. Apress (2007).
33. Segaran T., Evans, C., Taylor, J.: Programming the Semantic Web. O'Reilly Media (2009).
34. Governor, J., Hinchcliffe, D., Nicknull, D.: Web 2.0 Architectures: What Entrepreneurs and Information Architects Need to Know. O'Reilly (2009).