Paper at edix\open\web\webtasks.doc (currently 4900 words; 4933 with footnotes)

Supporting web development in the OPEN process: additional tasks

B. Haire, B. Henderson-Sellers[*] and D. Lowe
University of Technology, Sydney

Abstract

To support commercial-strength web development it is as important to utilize a process as it is in regular, non-web information systems development. Using an established OO/CBD development process (OPEN), we evaluate its efficacy in web development support and propose new and amended Activities and Tasks that should be included in OPEN in order to fully support the new demands of website construction and the delivery of business value on the web. Sixteen new Tasks are identified together with one new Activity. Four subtasks of particular relevance to the interface and based on Usage Centered Design are also advocated.

Keywords

Web development; object-oriented development; OPEN process; tasks

1. Characteristics of Web Development

It is often claimed that web development is inherently different from standard applications software development (e.g. Bieber and Isakowitz, 1995; Burdman, 1999; Overmyer, 2000). Yet web development in its current incarnation goes far beyond the "promotional brochures" and "eye candy" of the first generation of websites and is concomitant with normal software development in a business environment PLUS a number of issues relating to usability (users can rapidly switch to a competitor's site if your website is too arcane), bandwidth (high volume of concurrent users) and graphic artistry (at least in the field of B2C). Web pages are often read in much the same way as brochures, usually scanned for important information and rarely completely read by the user. Web development projects create forms of consumer media with videos, sound clips and sometimes entire movies. In addition to this, there is also the traditional software aspect to web development with web sites quite often containing sophisticated back-end systems that help sort, organise and maintain the site. Timescales for website development are also often short and site contents extremely malleable; web projects tend to be very visible in nature. Systems that face the outside world have no room for error. The consequences of errors and downtime in web systems that interface to customers or suppliers are often major and simply cannot be tolerated. This results in the need for systems and upgrades to be right first time every time. Possibly even more significant, from a development perspective, is the lack of

[*] Corresponding author: Professor B. Henderson-Sellers, Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia. Tel +61 2 9514 1687; fax +61 2 9514 1807, email: brian@it.uts.edu.au

certainty in the system domain and the volatility in the requirements of the system – which invariably evolve considerably as the system design emerges (Lowe, 2000). Indeed, for much commercial development, the requirements process can be viewed as design-driven requirements management. In other words, the design process is explicitly used to reduce requirements volatility. This has a fundamental impact on the overall process that is adopted.

1.1. Architecture

The architecture of a web project is extremely important to its long-term success. The architecture typically merges a number of separate aspects. Specifically, it covers both an information architecture and a technical architecture. The information architecture covers aspects such as the underlying content, the way this content is structured and managed, together with a link between the information and the business model that is being supported. The technical architecture of Web systems typically has a thin highly-customised client front-end, a substantially component-based middleware layer linked together with appropriate "glue" code and a customised back-end that links the system together with legacy systems.

One feature that must be highlighted in the project's architecture is that it must be adaptable. Technology within the web development field is changing so quickly that the architecture must be designed in such a way that it can easily change with technology. The importance of a project's architecture is often overlooked and many people assume that, since the system exists and has been built, then it must have an architecture. This is not true. Recently the re-use of system architectures within the web development community have been gaining support. Large companies such as IBM, Sun, HP and Microsoft have begun long-term projects that deal specifically with web projects and their architectures (Butler, 2000).

1.2. Component Based Development

The software industry is approaching a stage within its development where software packages, called components, can be used to assemble systems, similar to the way you would put electronic components together on a printed circuit board. Generally, some custom coding still needs to be completed in order for the components to interact with each other. These components are continually becoming more advanced and the amount of coding needed is becoming more limited. This component-based development is also very apparent within web development projects. Web developers can assemble applications using a combination of remote services and local services. The nature of such component-based development differs slightly from traditional OO development and so new processes for development must be sought. The adoption of a component-based approach in Web development is also reflected in the emerging Web design notations such as WebML (Ceri *et al*, 2000) and Conallen's work on the adaptation of the UML (Conallen, 2000).

1.3. Content Management

Central to the idea of web development is the idea of an information architecture. A key aspect of this is content management. The rate of change within traditional software projects does not compare with the rate of change within today's web projects. Web sites where the content is updated several times an hour are commonly found. Web site architectures must be adept at handling this ever-increasing rate of change within systems. This suggests there is a need to consider carefully the link between the information architecture and the technical architecture of a system. Typically, these are developed and represented using quite disparate approaches and the understanding of how to link these is only just beginning to emerge. Without this understanding, it becomes difficult to ensure that aspects such as the content management system are effectively embedded within the overall systems architecture.

1.4. Interfaces

The interface wars have now entered the web. The level of communication between projects is increasing as the Internet becomes faster and more reliable. Businesses are using the Internet to rid themselves of paper-based processes and to improve their systems. The emergence of a new power player, XML, within the Internet has emphasized the importance of a standardised communication language. The value of data is known and more time is being invested in its ability to adapt. Many organisations have been through the painful and expensive process of converting or interfacing legacy systems and do not want to repeat the process. This extra attention to the data within a system affects the process used when developing web projects.

1.5.Requirements Engineering and High Level Design

The development of prototypes or white sites seems to be common practice within a number of web development organisations. At a recent Object World 2000 conference in Sydney, a number of companies described producing white sites in the requirements elicitation phase of development, estimating the requirements phase to represent 15%-20% of the total effort expended on a project.

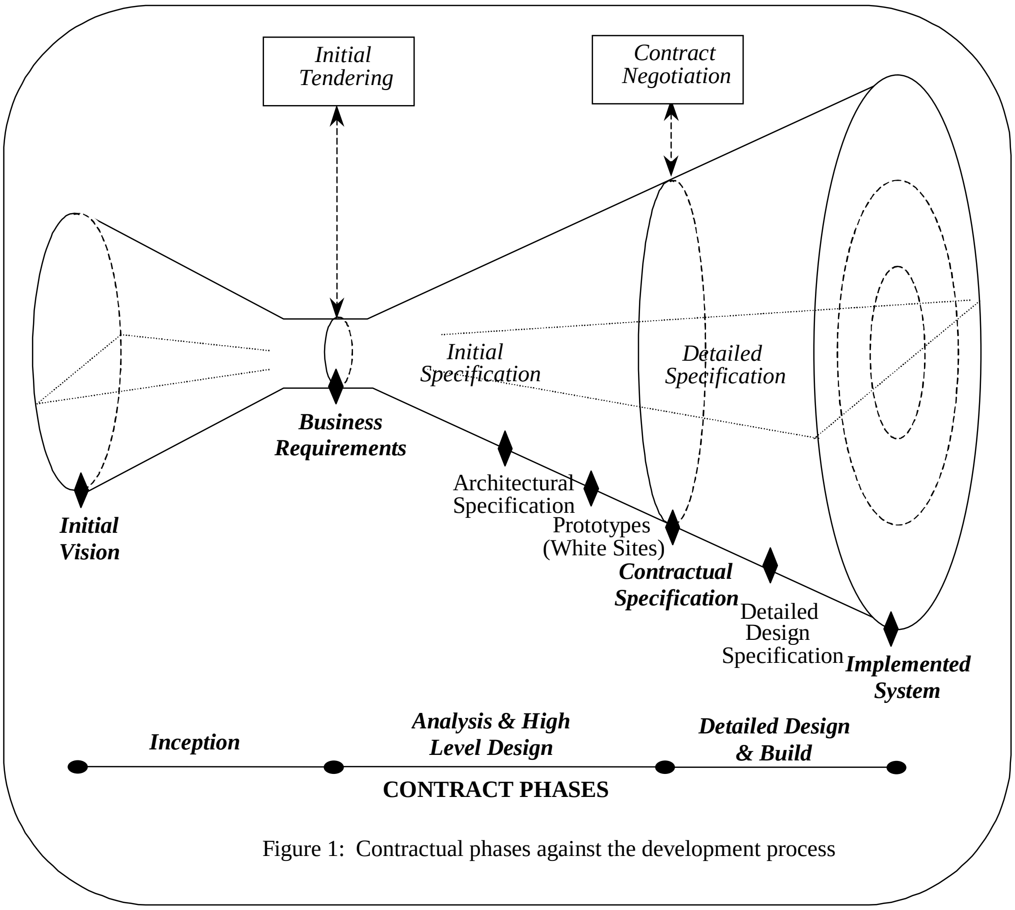

Figure 1: Contractual phases against the development process

What is important is that the architecture is developed during the requirements phase. This is the solution to the fact that requirements within web development projects are extremely volatile. This can often be because the client is unsure of what can be achieved, changing technology, or any range of situations. This is not a surprising attribute of any project, although it is more apparent in web development. Perhaps this is because the Internet is a relatively new technology, even within the context of software development. The production of a white site, and therefore the high level architecture, solves this problem by leaving the final contract of the systems specification to later in the development cycle. **Figure 1** depicts what happens in web development projects. What is interesting about this diagram is that there is no separate design phase that is presented to the customer. Rather it has been broken into two: high level design concerned with the architectural structure of the solution and lower level detailed design concerned with the design of the architectural modules. The first of these two, the high level architectural design, has been incorporated into the requirements finding or analysis phase of development. The latter of the two, detailed design, has been moved into the production or build phase of development. This makes the distinction between analysis and design hard to identify within web projects. The majority of analysis techniques used today in web development tend to point towards design decisions. Use cases are considered an analysis tool, yet there are a number of design decisions made while using them, as pointed out by Constantine (2000).

2. A Starting Point for a Web Development Methodology: OPEN

Object-oriented Process, Environment, and Notation (OPEN) is the longest established third-generation, public domain, full lifecycle, process-focussed, methodological approach that was designed for the development of software intensive applications, particularly object-oriented and component-based development (Graham *et al.*, 1997; Henderson-Sellers *et al.*, 1998).

OPEN is defined as a process framework, known as the OPF (OPEN Process Framework). This is a process metamodel from which can be generated an organizationally-specific process (instance). The major elements in this metamodel are Work Units (Activities, Tasks and Techniques), Work Products and Producers. Together these produce Work Products and the whole process is structured temporally by the use of Stages (phases, cycles etc.). Each process instance is created by choosing specific Activities, Tasks and Techniques and specific configurations thereof. OPEN thus provides a high degree of flexibility to the user organization.

The componentized nature (afforded by the metamodel) of OPEN thus permits the scope of the approach to be extended whenever new technologies arise – or rather, whenever the development context changes, thereby requiring changes to the development approach. Two such examples of changes are the emergence of component-based development and web engineering (although there is in fact significant overlap). Extensions to OPEN to support CBD are given in Henderson-Sellers (2001). In this paper, we concentrate on adding support for web-based developments, firstly by asking what support already exists in OPEN and then what is missing and therefore needs to be identified, described and defined for addition into the OPEN framework.

3. Existing OPEN support for Web Development

We have discussed the differences that exist between web development and traditional software development, but there is also a lot in common between the two fields. Therefore a lot of the activities, tasks and techniques in the OPEN framework, are still relevant to web development. Here, we evaluate existing, new and modified Activities and Tasks needed to create Web OPEN as a web-enabled "dialect" of OPEN.

If we consider the similarities between regular and web development at the granularity of OPEN's activities, the tasks relevant to the activities of *Project Initiatio*n, *Implementation Plannin*g and *Project Planning* will remain relatively unchanged. These activities and tasks are the same for any project. Business approval must be obtained, feasibility studies must be undertaken and other general tasks must be completed. Activities such as Requirements Engineering and build will be most affected, as this is where the project domain affects the process. Illustrated by **Figure 2** is OPEN's coverage of lifecycle issues. As OPEN is a full life cycle process model it takes into account business, training and personnel issues. The activities, tasks and techniques associated with these issues may vary, but will not be considered in this paper.. Rather we will focus on the technical aspects of the process.

There is no need to create a new set of activities and tasks to mimic the ones already there. Most of the activities, tasks, and techniques are generic enough to be used in web development, e.g. Code can be used to signify the coding of objects or the actual writing of HTML pages. Some activities become less critical in Web development, or

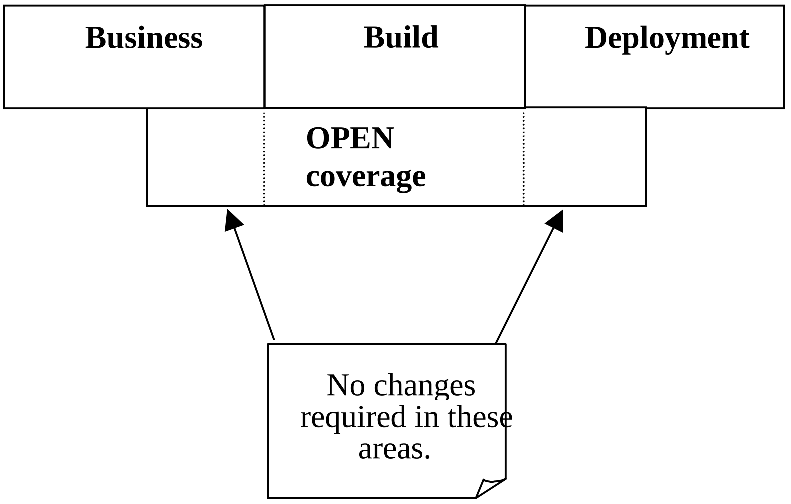

Figure 2 OPEN's coverage of lifecycle issues

more critical (such as configuration management (Dart, 2000)), but the *existence* of the activities is not affected. Changes could be made to make the activities, tasks, and techniques more specific to web development. For example *Undertake Architectural Design* could be renamed *Undertake Web Architectural Design* for a web environment; although since it does not really add value to the task, such a renaming is probably not necessary.

The tasks and techniques relating to the development of objects will still be useful within the Web OPEN framework. Their importance may be reduced, however, as a component-based development process becomes more prominent.

4. Extending OPEN support for Web Development

In this section we outline the various activities and tasks (**Table 1**) that are proposed as additions and modifications to the OPEN framework to better facilitate web projects (supporting techniques are to be discussed in a companion paper: Haire *et al*., 2001).

Table 1 Summary of new Activities and Tasks for Web OPEN

New Activity

Website Management

Tasks

Build White Site
Create content (on website)
Create navigation map for website
Define acceptance criteria for website
Define website testing strategy
Design and Implement content management strategy
Design and Implement personalization strategy
Design website architecture
Design website standards
Develop a brand identity
Develop Data Standard
Integrate Content with User Interface
Prototype the human interface
Undertake content management
Undertake market analysis
Undertake testing of website

Subtasks
Choose Architectural Pattern for website (subtask of "Create a System Architecture")
Create the UCD role model (subtask of “Design User Interface”)
Create the UCD task model (subtask of “Design User Interface”)
Create the UCD content model (subtask of “Design User Interface”)

Relevant CBD Tasks

Choose appropriate component framework
Evaluate the potential component frameworks
Integrate Components
Screen the candidate list of component frameworks

4.1 New Activity: Website Management

The whole area of web engineering requires a significant new focus in the form of an activity that is proposed as "Website Management". Website management brings together all the issues regarding the development, maintenance and management of a corporate website which may or may not include access to back-end transaction processing systems. The objectives of the website management activity include creating a high quality website; keeping the website up to date; and ensuring that site standards are met as the website evolves.

The website management activity involves a number of new OPEN Tasks which are introduced in the remaining subsections below, listed in Table 1 and described in detailon the OPEN website (http://www.open.org.au).

4.2 Components and Frameworks

Web projects tend to have at least one level of their architecture that is component based. The OPEN framework detailed in “The OPEN Process Specification” does not include adequate support for component-based development (CBD). There does, however, exist an extension to the OPEN framework that allows for CBD. This extension can be found in more detail in Henderson-Sellers (2001) and will not, therefore, be discussed any further here, while noting that the nature of web development projects, and their component-based architecture, means this extension would be useful when implementing a Web OPEN framework.

4.3. Content Management and Personalization

Another major factor within web development is both the idea of content management and personalization. These both represent functionality that must be included in the majority of web projects today. It can be debated as to the level of which these factors should be represented in the OPEN framework. Since these issues can be considered to play an intricate part in the architecture of the solution, new tasks are proposed for “Design and implement content management strategy”; and “Design and implement personalization strategy”. These tasks can often be component based and so could often run parallel to the new activity of “Component selection”.

4.4. Architectural Patterns

In recent times there has been much discussion on patterns within software development. This is also the case in the web development community. A number of large organisations have done a significant amount of work in detailing architectural patterns that emerge for different web applications. Some of these initiatives link in closely with their corresponding component frameworks (e.g. Sun’s initiative, Java 2 Platform, Enterprise Edition Blueprint focuses on the development and deployment of applications using the J2EE platform); however, others focus more on the architectural side of the solution and separate this from the implementation layer of development. The most notable of these is the work done by IBM in their “Patterns for e-business” project (Butler, 2000). This all points to the need for choosing an architectural pattern (or framework if you like) depending on the type of web development. These architectural patterns are really a form of domain modeling in the OPEN framework. To facilitate this choice within the OPEN framework a new task “Choose architectural pattern for website” has been created.

4.5. Content Development

Within web projects there is a large amount of rich content that exists as part of the user interface. In traditional software development, the user interface contained a number of various simple controls such as combo boxes and edit bars. The user interface with web development projects consists of almost anything imaginable, from rich text, streaming audio, to even actual applications within the user interface. This content must be carefully prepared much the same way that an editor reviews a newspaper before its final print. The preparation of images, editing and layout of text, and obtaining copyright clearances all must be completed. There is no existing task that deals with these kinds of things within the standard OPEN framework. The addition of such a task is essential for a Web OPEN framework. The name chosen for this new OPEN task is “Create content (on website)”.

After this content has been carefully prepared it must then be combined with the user interface. This is an ongoing task that must be done to bring together the worlds of print media and software development. On one side you have a team of creative type people coming up with all sorts of new ideas. On the other side you have your technical people who must facilitate a method to integrate these ideas with the current navigation, usage and content management of the site. A new OPEN Task of “Integrate content with user interface” is thus proposed. This task is responsible for combining the content with the method being used to present that content to the user. This task is also important within the Web OPEN framework as it highlights the difficulties that occur when combining two different cultures together within the same project.

4.6. User Interface

The user interface within a web project constitutes a large portion of the overall project. It is vital in determining the success or failure of the project. OPEN already has a task named “Design user interface”. This task needs to be somewhat more emphasized for web development projects. It does not warrant being labeled as an activity under the OPEN framework, yet deserves a number of relevant tasks. These tasks have been taken from Constantine and Lockwood’s (1999) work on Usage Centered Design (UCD), which is more appropriate than the significantly different User Centered/Centric Design (Norman and Draper, 1986) given that it is often not possible to conduct effective user centred design. “Usage-Centered Design focuses on the work that users are trying to accomplish and on what the software will need to supply via the user interface to help them accomplish it.”(Constantine and Lockwood, 1999). It is also important to recall the comments made in the introduction about the role of design within Web development. In particular, design-driven requirement elicitation is significantly different from conventional design. This highlights the significance of UCD, which allows designers to focus on potential patterns of utilisation and therefore helps resolve the uncertainty in the requirements.

The three sub-tasks that have been added to supplement the original “Design user interface” task are “Create the UCD role model”; “Create the UCD task model”; and “Create the UCD content model”. The last of these subtasks links in well with the new task “Integrate content with user interface” as it starts to identify the relationships between the content and the user interface including navigation maps (new Task: "Create navigation map for website"). All three subasks identify how the site is to be used (hence the name Usage Centered Design) and also help to tie the user interface to the web projects requirements.

4.7. White Sites (the web prototype)

A task often completed in web development projects is the construction of what is termed in the industry as a “white site”. A white site consists of no rich graphical content and usually only represents a portion of the entire web site. In web development, the white site is responsible for a number of key factors that lead to a successful project:

- it provides valuable client confirmation that the developer has interpreted the systems requirements correctly;
- it integrates the change management strategy, the usage and the navigation of the site all together in a visible working solution;
- it works as a communication tool to display the site architecture to the client and the development team.

Strictly speaking, creating a white site is a form of prototyping, which is already covered in the OPEN framework. However, the usefulness of white sites, in the design and requirements engineering phases of web development, warrants the existence of a separate task. In addition, we notethat there exist a number of various techniques relating to how to create white sites, whichpoints to it being a task rather than a technique. The name chosen to represent this new task in OPEN is “Build white site”.

4.8. Interfaces

At present the rapid pace of technological change and the growing complexity of Web systems is leading to significant difficulties with regard to interfaces between various system components and between Web systems and both legacy systems and related business systems. Although the technology (and in particular the communications protocols and data formats) supporting these interfaces will stabilise – led in part by the move to XML – other aspects will continue to evolve. As interfaces stabilise, changing knowledge representations will become a major focus, then as these stabilise changes in agent brokers may become the focus. In other words, changing technology has become a constant factor within the Web environment. Nevertheless, the development of data standards is still a critical aspect of Web development, given the strong focus on content and the way in which it is managed. Consequently, an appropriate new OPEN Task is introduced: “Develop data standard”. There has been a lot of work done on developing industry standard XML tags by the w3 consortium and this could often provide an excellent starting point for this task. The importance of this task is more noticeable in large B2B projects than B2C projects.

4.9. Performance Testing

Performance testing has been elevated to a new level on the web. We now see systems that must deal with tens of thousands of concurrent users on-line. The performance of a web project can often be determined by its ability to deal with these large loads. This links into the fact that users can quickly become frustrated from an unresponsive web site and quickly move on to another. The OPEN framework does not currently have a task relating to this kind of performance testing. The inclusion of a new task “Undertake testing of website" highlights the importance of performance testing within web development projects. (Tasks and Techniques already exist in OPEN for usability and interface testing and evaluation e.g. Technique: Usability testing.)

4.10. Market Research

There are many tasks that are borderline between being classified as dealing with the engineering side of software development and the business side of software development. In web development this tends to be somewhat more evident. A new OPEN Task of "Undertake market analysis" is heavily associated with the upper business level, since there must be some understanding of the target audience before the organization decides to embark on a web development project. Unfortunately, with the state of affairs recently, this does not seem to be the case in practice. Many organisations simply want a web site, almost for its own sake or "for appearances", and they will find out who they are targeting later. Also there is a significant amount of information that web developers need to know about their target audience with which the upper business levels would not be concerned. For example, an initial market research project may cover the disposable income of web users, their age range etc. Web developers will want to know quite different things such as What system are they using? How fast is their connection? The importance of this task is more noticeable in web projects that fall into categories that deal with customers (e.g. B2C). It is still important in other web projects (such as B2B, or B2E) but the information is generally easier to obtain and does not always warrant a separate task.

5. Summary

5.1 Conclusions

Web projects must be developed with the emphasis being on how services can be improved and not on the technology involved. Success on the Internet is more than having the fastest computers and the biggest databases; it is about complete fulfilment of the business processes. This includes issues such as customer query handling, product delivery and tracking, as well as service guarantees and speed of connections. It is also important to note that these business processes themselves are changed by the introduction of the new systems – so we end up with a system that by its very nature will modify its context and thereby require further changes in order to remain effective.

This emphasis is reflected in Web OPEN with the addition of tasks relating to the analysis of the data and how they will be used within the final system. This emphasis also needs to be represented within the user interface for successful web projects. This has been completed in Web OPEN by the inclusion of usage-centered design tasks and techniques (Constantine and Lockwood, 1999). Usage-centered design focuses on how the user interface will be used in order to improve the underlying business need of the system, rather than on who will be using it as in user-centered design.

A number of other smaller modifications have been made to the OPEN framework that are particular to web development, including testing, market analysis, and the development and review of content. OPEN's strength in the field of metrics is maintained with the addition of tasks relating specifically to web metrics.

5.2 Further Work

Web OPEN is a starting point for further research into web process development. Further work could be completed on the task *Integrate components*. The importance of this task will continue to escalate, as components become more developed and widely used. A significant amount of work on component integration has been completed by the Catalysis team (D'Souza and Wills, 1998) and would prove beneficial to the Web OPEN framework. Component integration is complemented by component creation. There is no reason why organisations cannot produce their own custom components and include this within their development process. Further work towards merging, with OPEN, ideas from usage-centered design is also under way.

The metrics that are specific to the web development process need to be further looked at, evaluated and statistically verified. Much the same way where good quality objects were shown to have less than 10 methods and 3-4 lines of code per method (Haynes and Henderson-Sellers, 1996), similar web metrics must be obtained. Once these appropriate metrics have been produced, we will be able to better measure the quality of web projects.

There will always be constant need for review and modification of the Web OPEN framework as technology and the development environment changes. Further work could also incorporate varying development methodologies such as artificial intelligence or agent technologies.

References

Bieber, M. and Isakowitz, T., 1995, "Designing Hypermedia Applications", *CACM*, **38(8)**, 26-29

Burdman, J., 1999, *Collaborative Web Development: Strategies and Best Practices for Web Teams*, Addison-Wesley

Butler, M., 2000, "IBM's Patterns for e-business", White-paper presented at Object World 2000.

Ceri, S., Fraternali, P. and Bongio, A., 2000, "Web Modeling Language (WebML): a modeling language for designing Web sites", in *Proceedings of WWW9 Conference*, Amsterdam, May 2000

Conallen, J., 2000, *Building Web Applications with UML*, Addison-Wesley, 300pp

Constantine, L., 2000, "What Do Users Want? Engineering Usability into Software", *Windows Tech Journal*, December 1995.

Constantine, L.L. and Lockwood, A.D., 1999, *Software For Use*, Addison-Wesley, 579pp

Dart, S. 2000, *Configuration Management: The Missing Link in Web Engineering*, Artech House,

D'Souza, D.F. and Wills A.C., 1998, *Objects, components, and frameworks with UML: the Catalysis approach*, Addison-Wesley, 785pp

Graham, I., and Henderson-Sellers, B., and Younessi, H., 1997, *The OPEN Process Specification*, Addison-Wesley, 314pp

Haire, B., Lowe, D. and Henderson-Sellers, B., 2001, "Supporting web development in OPEN: additional techniques and roles", submitted for publication

Haynes, P. and Henderson-Sellers, B., 1996, "Cost estimation of OO projects: empirical observations, practical applications", *American Programmer*, **9(7),** 35-41

Henderson-Sellers, B., 2001, "An OPEN process for component-based development", invited chapter (Chapter 23) for *Component-Based Software Engineering: Putting the Pieces Together* (eds. G.T. Heineman and W. Councill), Addison-Wesley, Reading, MA, USA

Henderson-Sellers, B., Simons, A.J.H. and Younessi, H., 1998, *The OPEN Toolbox of Techniques* Addison-Wesley, UK, 426pp + CD

Lowe D, 2000,"A Framework for Defining Acceptance Criteria for Web Development Projects", Second ICSE Workshop on Web Engineering, 4 and 5 June 2000; Limerick, Ireland

Norman, D. A. and Draper, S.W., 1986, *User-Centered Design*, Hillsdale, N.J.:Lawrence Erlbaum

Overmyer:, S. 2000, "What's Different about Requirements Engineering for Web Sites?", *Requirements Engineering Journal*, **5**(1), 62-65

Wirfs-Brock, R., 1993, "Designing Scenarios: Making the Case for a Use Case Framework", *Smalltalk Report*, November-December