Adding Web Support to OPEN

B. Henderson-Sellers, B. Haire and D. Lowe

Printed: March 16, 2001

for ROAD/JOOP June 2001

Deadline March 19

Address for Correspondence
Professor B. Henderson-Sellers
Director, Centre for Object Technology Applications and Research
Faculty of Information Technology
University of Technology, Sydney
P.O.Box 123
Broadway
NSW 2007
AUSTRALIA
fax: +61 2 9514 1807
email: brian@it.uts.edu.au
Today's object-oriented processes and methodologies have often evolved over a significant number of years. Commercial web development is a more recent phenomenon. Consequently, publications defining object-oriented and component-based (OO/CBD) processes often have little to say in their standard, published versions about how to use them in a web environment.

In the OPEN community, this deficiency has been recognized and recent work has identified necessary extensions and modifications to the published work on OPEN Activities and Tasks\(^1\) and OPEN Techniques\(^2\). To complement the OPEN extensions to component-based development discussed in my last column\(^3\), here we consider what additional Activities, Tasks and Techniques might be added to the OPEN repository to more fully support the needs of today's web developers.

**WEB DEVELOPMENT**

It is often claimed that web development is inherently different from standard applications software development. 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 read completely 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 websites quite often containing sophisticated back-end systems that help sort, organize 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 certainty in the system domain and the volatility in the requirements of the system which invariably evolve considerably as the system design emerges. 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.

**EXISTING OPEN SUPPORT FOR WEB DEVELOPMENT**

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 Initiation, Implementation Planning 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.

There is no need to create a new set of web-based activities and tasks to mimic the ones already in OPEN. Most of the activities, tasks, and techniques are generic enough to
be used in web development. For example, Task: 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, with others such as conﬁguration management becoming more critical; in other words, the existence of the activities is not affected, only the emphasis.

EXTENDING THE SUPPORT

In extending OPEN to more fully support web development, only one new activity has been identiﬁed, together with a large number of new tasks and a few techniques. These are all outlined below in alphabetical order. Also included is a brief description of some of the new roles required for members of the web development team.

New Activity: Website Management

Since the management of website development has a number of speciﬁc differences from conventional development (and is therefore not covered by any existing OPEN Activities), it is important that a new Activity be introduced: Website management. As one example of the differences, the management of Web systems development will typically need to manage (much more so than for conventional development) the balance between creative designers and technical developers. This new activity focusses on all aspects of web development, both technical and administrative, the overarching goal being to create and maintain a viable, eﬀective and e±cient website with all the required business functionality.

New Task: Build White Site

Since internet technology is constantly changing, it is often the case that either the client or developer (or both) are unfamiliar with the technology at hand. The production of a prototype allows for increased estimation and risk assessment for the remainder of the
The prototype most often developed in the web industry consists of a series of web pages with rudimentary content (graphical imagery is usually left out or remains in a very early stage of development) that sits on top of a simulation of the final architectural design. This type of industrial prototype has been labelled a "white site".

A white site incorporates all of the major architectural components of the design without worrying about the lower level details. It typically would include a rough change management solution (or at least an example of how it would work) and indications of the functionality to be supported, as well as sample pages of content.

New Task: Create content (on website)

The purpose of this task is to finalize the content that will eventually be incorporated into the user interface. It parallels what an editor might do in the production of any print media. Copyright clearances should be obtained for any hypermedia information to be displayed. The content should be edited to maintain a consistent feel. The overall relevance of the information should be reviewed for fitness for purpose.

New Task: Create navigation map for website

Many of the earliest websites just grew without any planning. One immediate consequence is that the user rapidly gets lost and disoriented when using the site because there is no navigational logic to the site. In this task, we purposefully design the content decomposition and the associated navigational structure of the site. This is often accessible these days via a button labelled something like "site map".

New Task: Define acceptance criteria for website

Websites are often created for the client by third party companies. In delivering the completed website to the client, it is important to know on what basis it will or will not
be deemed acceptable. This task focusses on agreeing the acceptance criteria early in the creation of the website so that they are unambiguous and easy to interpret. They can also form part of the assistance to the designer since he/she knows on what basis the site will be accepted.

New Task: Define website testing strategy

Unlike most (though certainly not all) conventional software systems, web systems are usually directly accessible to users from outside the client organisation. Indeed they often become the primary interface between the client and their customers. Hence, it is critical that the system be fully operational from initial release and not suffer performance or usability problems. This will often be complicated by the fact that the user base is only poorly known and potentially huge, as well as from diverse social and cultural backgrounds. This means there will be a multitude of assumptions in the users' minds as they use the site; and many unforeseen navigational paths through the site are likely to be realized. It is therefore critically important to thoroughly test the website before release. The testing must be rigorous and as complete as feasible. To do that, standards for testing must be pre-determined and an overall testing strategy devised to parallel the strategy created for non-website developments.

New Task: Design and Implement content management strategy

The power of web application and web projects lies in the ability of users to be able to access up-to-date information quickly. Without a mechanism for keeping the information within a web project up-to-date this advantage is lost.

The purpose of this task is to design and incorporate a system to allow the content within a project to be updated both easily and quickly. The extent to which this affects
the overall architecture of the solution depends on the type and extent of content to be updated and the method used to update it.

New Task: Design and Implement personalization strategy

As web projects can often deal with a large range of target users, it makes the task of designing a usable interface difficult. Each user has different expectations about how they will use the system, including both the interface and the functionality that will support them in achieving their goals. To overcome this problem the notion of user adaptation or personalization has been created. Details about the user (or class of users) are stored and the system adapts the content, structure, presentation and/or functionality to suit the specific user.

This task looks at defining the level of personalization allowed for the system. It looks at the type of user profile information that will be stored and the method used to do so. The extent to which this affects the overall architecture of the solution depends on the type and extent of personalization and the method used.

New Task: Design website architecture

Websites tend to evolve constantly during their lifecycle. This evolution is often very incremental and fine-grained. Without a solid architecture, it is likely that this evolution will result in a system that rapidly deteriorates in quality. This task focuses on creating such an architecture including considerations of security, usability and functionality. While websites are continually changing as owners update both content and functionality, it is important to permit this flexibility within an overall strategy (the architecture).

It is also worth noting that the architecture needs to encompass not only the technical structure, but also the information architecture | how the content will be managed,
organized etc. At present, architectural modelling approaches do not handle well the integration of these various aspects.

**New Task: Design website standards**

Users like a website that has consistency within the website and consistency with other websites; consistent user functionality such as they might expect from buttons, scroll bars etc. both in terms of their location and functionality. Since there are no internationally recognized standards for websites, it is important that a website owner consolidate a set of site-specific standards and stick with them. That way users can come to know what to expect and thus are more likely to keep returning to the site.

**New Task: Develop a brand identity**

Branding can often be highly lucrative. A product with a brand can sell for significantly more than an equivalent unbranded product (and vice versa). In terms of websites, branding the products and the website so that it is immediately recognizable worldwide takes patience but is an important part of many dot.coms these days. Websites such as www.amazon.com have entered the general parlance and thus can be counted as successful branding forays.

A subsidiary aspect of this task is the adoption of suitable meta-data standards and formats, as well as procedures for ensuring their maintenance. Meta-data can be critical in supporting effective indexing, searching and control and management of the data.

**New Task: Develop Data Standard**

The importance of the data within a project cannot be underestimated. The problems apparent with many of today's systems result from old legacy databases that are not adaptable enough to be able to handle changes and modifications. We cannot afford to
make this mistake in the future and so the data collected and used within web projects must be extensible, scalable and preferably follow some type of standard in order to be compatible. There are industry standards that are being developed that can act as an excellent starting point. Some of these include the W3 consortium's industry standard for XML tags. A well thought out data standard ensures that the system can be replaced with changing technology without affecting the data within the system.

**New Task: Integrate Content with User Interface**

This is the task of bringing two varying worlds of different disciplines together. The creative design team involves graphical designers and editors, whereas the software engineering team consists of web developers, software architects and programmers.

During this task an agreement must be reached by both teams as to the content that is to be displayed and the method used to display it. In effect, there will be a trade-off between what is desirable from a creative perspective and what is technically feasible.

**New Task: Prototype the human interface**

Since the value of most websites is in terms of attracting users and then getting them to return to the same website later, usability is a key issue for website designers. The human interface is thus critical and, perhaps as part of building a white site, an important task is to trial or prototype the human interface.

**New Task: Undertake content management**

Content on a website can change rapidly, sometimes daily. Managing this content is a change management challenge. Because pages are hyperlinked together and often to other sites (to and from) nothing is more frustrating than to try to follow a link that
ends in a cul-de-sac or content that is out of date or contains invalid links. Good content management would obviate such problems.

New Task: Undertake market analysis

The purpose of this task is to determine market share and sector characteristics. Market research is a huge industry and, if any significant research needs to be undertaken, it is recommended to outsource this particular task.

This task is useful in requirements engineering as it helps identify the usage of the system. It should provide information that better refines the requirements as well as affecting the overall design of the system. Things that should be looked into include average hardware profile of users, software most used for browsing, as well as non-technical issues such as the most common reason users would visit the site.

New Task: Undertake testing of website

This subtask has a two-fold purpose. The first is to test the hardware configuration and its ability to handle appropriate (possibly high volume) traffic or "hits" as they are commonly referred to. The second and perhaps more important aspect of this subtask is to test the underlying processes and how they are affected by high volume demand. There is no point in having a e-commerce web project that can handle 1 million hits a day if the fulfillment procedure responsible for delivering goods can only handle 100,000 orders a day.

New subtask: Choose Architectural Pattern for website (subtask of Create a System Architecture)

The purpose of this task is to minimize design time and risk by selecting an architectural pattern that has already been developed and tested. This pattern can then act as a starting point for further development. An unfortunate problem with selecting an
architecture is that it is often restricted by the choice of a component framework. For example, it would be more time consuming to implement anything else but a Microsoft architecture on a .NET framework. It is for this reason that it is recommended that this task runs parallel with the task “Choose Appropriate Component Framework”.

New subtask: Create the UCD rôle model (subtask of Design User Interface)

The purpose of this task, taken from Usage-Centered Design (UCD), is to identify whom, in terms of rôles, will be using the system. A user rôle is an abstract collection of needs, interests, expectations, behaviours and responsibilities characterizing a relationship between a class or kind of users and a system. User rôles generally relate to humans or more specifically to the rôles that the humans assume while using the system.

Some appropriate questions to consider while constructing the rôle model are given by Constantine and Lockwood. In addition, the development of the user rôle model also includes identifying focal rôles and creating a user rôle map.

New subtask: Create the UCD task model (subtask of Design User Interface)

The purpose of this task is to identify the nature of the work that is to be completed by the system. OPEN techniques such as videotaping are useful for this task since users will often tell you what they are meant to be doing rather than what they are actually doing.

The creation of essential use cases and a use case map are central to this task. Constantine and Lockwood (Chapter 5) present significant further detail on the process of creating a UCD task model.

New subtask: Create the UCD content model (subtask of Design User Interface)
In order for a system to be effective and user friendly, the appropriate tools and information must be available in the correct areas. The identification of these areas is dependent on the roles the user is playing (the role model) and the type of work that is being performed (the task model). The content model is an abstract representation of the contents of the various interaction spaces for a system and their interconnections (page 126). To communicate this, a content model with supporting navigation maps is used.

Other tasks

In addition to these new web-focussed tasks, some of the new CBD Tasks are also useful. These (simply listed here) are:

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

New Technique: Branding

Web development is increasingly less about software development and more about marketing and developing a market identity or brand. This is a combination of the product and the way that the product is portrayed on the website (and other media). Brand strategies must be developed, an overall artistic and marketing agreement made as to what constitutes the "brand" and ways to attain widespread brand recognition. An example here in early web commerce was www.amazon.com.

New Technique: Development Spikes

A development spike can be thought of as research. The aim of the technique is to minimize areas of high risk by diving right in and starting development within a certain
technology. The idea is that, once development has begun, a greater understanding of
the problem will be obtained and risk and time assessment will be more accurate. The
development done within development spikes can be used as a reference but should never
be used in the actual final development of the project.

New Technique: Field Trip

This technique is really quite self descriptive. It serves the same purpose as school
field trips or field trips in the natural sciences or the engineering professions. By actually
visiting a site, a greater overall understanding of the problem is gained. This technique is
useful in isolating implied (or assumed) user requirements.

New Technique: Reuse of Graphical Components

This technique encapsulates good practice, being based around the concept that
browsers cache pictures and therefore reusing a number of pictures will improve a site’s
performance and hence its quality. This technique also focuses on minimizing the size of
graphics without losing a significant amount of picture quality.

New Technique: System Metaphors

A technique originally used in Extreme Programming (XP) for naming classes and
methods, the use of system metaphors aims to keep the entire team thinking along the
same lines when it comes to naming. However, this is also an important technique when
discussing an architecture of a system. Since web development teams tend to contain a
wide range of producers from varying disciplines, the use of a system metaphor is a good
tool to communicate across these platforms.

New Technique: Web Metrics
One of OPEN’s strong characteristics is its attention to metrics. While further work is needed in order to statistically verify what the most appropriate metrics are, an initial proposal should focus on:

² Interface complexity: at a simple level this can be estimated through a page count, though this can be misleading, as a single server-side page may contain scripting that results in many different client-side manifestations. A more effective metric may be interaction counts or something similar, though there has, to date, been little work in this area.

² Performance: this can be estimated initially through number of hits per page per unit time | determines the general usage of a page and indicates where optimization would best be served.

² Access: Total size of pages (including graphics) | a useful quality measure in terms of speed to load.

² Maintenance: Rate of change of content | a metric useful for indicating when a site or page has become stagnant.

Some of the metrics are relevant to the development process and some related to the maintenance aspect of web projects. In addition to these extra metrics supporting web development, the original metric techniques within the OPEN framework are still relevant.

New Technique: Web Templates

This technique focuses on isolating common areas of content so they can be displayed in a consistent format. It also assists in the maintenance of a system by providing appropriate templates to use when adding new content to the site. Templates generally come with some kind of validation system that verifies that all the appropriate content has
been entered. Some commonly used methods include Microsoft Word™ documents with embedded Visual Basic to validate the content, or online Active Server Pages (ASP) that allow administrators to update and validate new content online.

The main advantage of using web templates is that it allows people with a lower level of technical experience with the system to do the majority of the work to create new content. This reduces the effect of any bottleneck that may occur with the site administrator. Web templates can also ensure that standards are met by forcing the collection of information such as metadata for indexing.

Rôles

Along with regular software development rôles such as Requirements modeller, System architect, System modeller, System/site administrator, System tester, Project manager and Prototype developer (now with a web focus) come also new rôles: of Web designer, Graphic designer and, perhaps, Editor.

Rôle: Web designer

The web designer needs to have a general skill level in a wide variety of areas. Ideally, they have some artistic ability for creating things such as simple graphics as well as general programming experience. The web designer helps to bind the gap between the artistic world of print media and the programming world of software engineering. They work with the graphic designer as well as the system developers to make the proposed content and layout a reality. Skills in HTML, Java Applets, Javascript, XML and many other web-based technologies are a necessity.

Rôle: Graphic designer
Due to the amount of rich content that goes into many web projects there is a need for this role of a graphic designer. Their responsibility is to help prepare the content and layout for the final system. This can include photographs, music clips, video clips and much more. The graphic designer needs to be artistic and imaginative and possess strong skills in creating and altering computer media.

Rôle: Editor (optional)

This is an optional rôle for many web projects. It is often left as the responsibility of the client to ensure that the content provided has been correctly edited and reviewed. In many cases, this rôle is not explicitly named but the tasks associated with the rôle are divided amongst the team, in particular between the web designer and graphic designer. In projects with a large amount of content, it is a good idea to assign someone to this rôle more permanently.

SUMMARY

With the increasing use of OO and CBD techniques in web development and the consequent use of an appropriate process, it is important that such a process offers full web support | here described in the context of one speciﬁc process: OPEN. Since the OPEN process itself is componentized, we have shown here how, with relative ease, OPEN has been complemented and extended by a number of Activities, Tasks and Techniques targetted for web developments.

Acknowledgements

This is Contribution no 01/06 of the Centre for Object Technology Applications and Research (COTAR).
References


8. Beck, K., 2000, Extreme Programming Explained, Addison-Wesley, Reading, MA, USA, 190pp
Authors' bios

Brian Henderson-Sellers is Professor of Information Systems in the Faculty of Information Technology, University of Technology, Sydney (UTS), New South Wales, Australia. He may be contacted at brian@it.uts.edu.au

Brendan Haire recently completed his Bachelor of Engineering (Computer Systems) degree in the Faculty of Engineering at the University of Technology, Sydney. He may be contacted at Brendan@sta@usyd.edu.au

David Lowe is Associate Professor in the Faculty of Engineering, University of Technology, Sydney (UTS), New South Wales, Australia. He may be contacted at david.lowe@uts.edu.au