

# Characterisation Of Web Projects

David Lowe [\[HREF1\]](#), Faculty of Engineering [\[HREF2\]](#), University of Technology, Sydney [\[HREF3\]](#).  
[david.lowe@uts.edu.au](mailto:david.lowe@uts.edu.au)

## Abstract

Both the research literature and commercial practice is increasingly acknowledging that Web systems development has unique characteristics, primarily reflecting specific elements of the systems being developed. One of the most significant of these differences is the way in which an understanding of client needs is developed and the systems are specified. Specifically, commercial practice tends to adopt an iterative process that carries out the initial specification by exploring with the client potential solutions (using techniques such as white sites). This serves a dual purpose of improving clients' understanding of their own needs as well as beginning the design process. The result is a specification that starts as a set of acceptance criteria and initially evolves into an architectural specification that captures client requirements, and then develops into a build specification (often embedded into the actual built system).

In this paper we present a model of these system characteristics, and show how they evolve over the lifetime of a project. This model is based on data collected from a series of interviews and surveys of commercial developers, as well as the analysis of commercial specifications. We show how this model can be used as the basis for guiding the development process, indicating the inter-relationships between the various emerging characteristics.

## Introduction

The rapid and successful deployment of Web-based systems has become increasingly important to the business strategy of many organisations. This is particularly true with respect to the way in which they interact with customers, clients, and/or business partners.

Our understanding of how to effectively develop these systems is continuing to develop. In particular, numerous development processes (see, for example ([Angelique, 1999](#); [Burdman, 1999](#); [Thomas, 2000](#))) and design methods (see, for example ([Baumeister, Koch, & Mandel, 1999](#); [Ceri, Fraternali, & Bongio, 2000](#); [De Troyer & Leune, 1997](#); [Erskine, Carter-Tod, & Burton, 1997](#); [Takahashi & Liang, 1997](#))) have been emerging in the literature. For an overview of this work as it relates to the early stages of the development process see ([Lowe & Elliott, Submitted](#)). A key observation in much of this literature is that Web systems have various unique features that impact on the approach we take to development ([Burdman, 1999](#); [Lowe & Henderson-Sellers, 2001b](#); [Overmyer, 2000](#)).

One of the most significant of these features is the level of client uncertainty regarding the problem domain and its relationship to possible solutions. This is typically handled in best practice commercial development by adopting an iterative process that initially explores partial solutions with the client (using techniques such as white sites, story boards, etc). This serves a dual purpose of improving clients' understanding of their own needs as well as beginning the design process. The result is an evolving specification that starts as a set of acceptance criteria and initially evolves into an architectural specification that captures client requirements, and then develops into a build specification (often embedded into the actual built system).

In previous work we have undertaken an extensive set of industry interviews and surveys. The interviews were intended to identify general perceptions and qualitative trends related to the process by which Web systems are specified. The surveys were intended to capture more quantitative information. The goal of both was to understand current best-practice, and the various motivations driving this practice. This work has resulted in both descriptions of the issues that are specific to the development (and in particular, specification) of Web systems and the identification of the elements that should be identified and explored during the initial specification of a Web system ([Lowe & Eklund, 2001](#)).

In this paper we extend this work, presenting a formal model of the characteristics of a Web system, and the way in which these characteristics are interrelated to each other, and related to the development process stages. In the following section we consider the background to this work, looking at the process by which Web systems are developed and how this impacts on the modelling of Web systems. We then describe the characterisation model that we have developed and show how this can be utilised in improving the development process. This is then followed by a consideration of examples of specific requirements, as identified from a series of analyses of commercial projects. We finish by considering further work and possible future directions.

## Background

As stated in the introduction, there has been an increasing awareness within the research literature that

Web systems have some specific characteristics that potentially impact on the development approach that is adopted ([Burdman, 1999](#); [Lowe & Henderson-Sellers, 2001b](#); [Overmyer, 2000](#)). Indeed, these characteristics have typically been much more readily recognised and addressed within commercial practice ([Lowe & Eklund, 2001](#)).

Research work has tended to focus on specific design methods - emphasizing either information and navigational design (e.g. ([Ceri et al., 2000](#); [Schwabe & Rossi, 1995](#))), functional architectures (e.g. ([Conallen, 1999](#); [Gellersen, Wicke, & Gaedke, 1997](#); [Russell, 2000](#))), or business patterns and how these relate to potential architecture (e.g. the work on patterns for e-Business work being developed by IBM ([Lord, 2000](#))).

There has been much less work that has investigated the broader development process. The research that does exist tends to be relatively prescriptive, and focus on management of the technological infrastructure (e.g. ([Angelique, 1999](#); [Burdman, 1999](#); [Haggard, 1998](#); [Thomas, 2000](#))). One exception to this is the work by the authors on extensions to the OPEN (Object Process, Environment and Notation) process framework ([Graham, Henderson-Sellers, & Younessi, 1997](#)) to make it more suitable for supporting Web development process. In particular, this work has identified additional tasks, techniques and roles that are required to support Web development ([Haire, Henderson-Sellers, & Lowe, 2001](#); [Henderson-Sellers, Haire, & Lowe, 2001](#)). Although this work provides a framework for describing Web development processes, it does not provide insights into the actual structure of processes as they are (or should be) carried out.

In this context we undertook a substantial series of industry surveys and interviews, coupled with the analysis of numerous Web specifications. The results of this work have been published elsewhere ([Lowe & Eklund, 2001](#); [Lowe & Henderson-Sellers, 2001b](#), [2001c](#)).

This earlier work highlighted that most commercial Web development is highly iterative, and that the process emphasizes the exploratory evolution of a system specification. The development can be viewed as following a dual-cycle process as shown in Figure 1. The first cycle iterates around a series of white sites, story-boards and other similar exploratory design prototypes, with the aim of moving from an initial set of acceptance criteria to a clear specification of the system - but a specification that includes not only requirements but also the broad architectural design elements of the site ([Gates, 2001](#); [Haggard, 1998](#)). The second cycle covers the (usually fine-grained incremental) design and build process. This second cycle (and indeed elements of the first cycle) bears similarity to lightweight incremental processes like XP (eXtreme Programming ([Beck, 1999](#))).

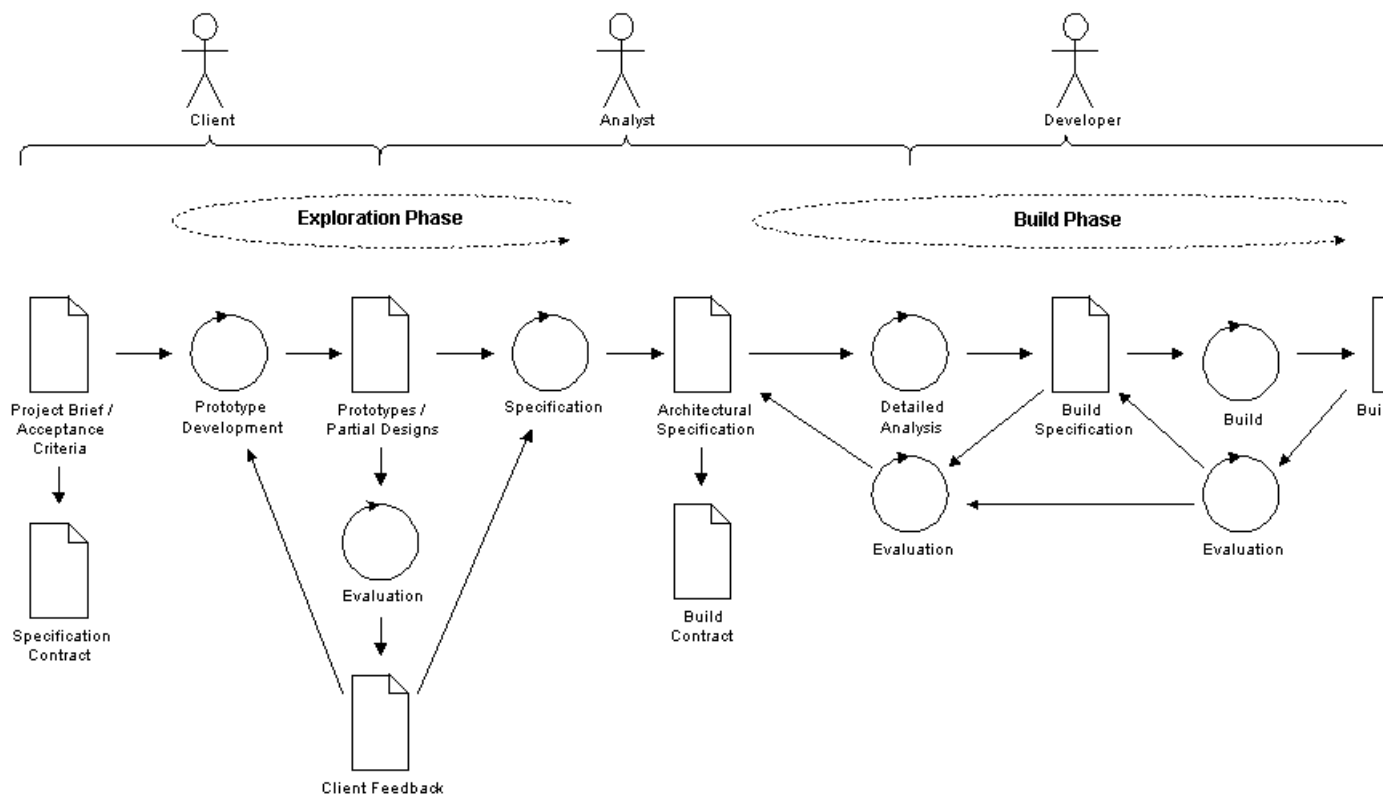


Figure 1: *Typical Web development process*

Care however needs to be taken in assuming that the process is simply a variant of iterative or incremental development. The lightweight and iterative processes that are adopted in conventional IT

development often use intermediate designs or solutions to obtain client feedback as a way to clarify their requirements. Typically, it is assumed that the client understands and is able to articulate their needs. This is much less true for Web projects. When applied to Web development, these incremental processes have a slightly different focus ([Angelique, 1999](#); [Fournier, 1999](#)) - supporting the development of problem domain understanding. In effect, the process (specifically the first of the two key cycles shown in Figure 1) is aimed at developing (or rather evolving) a joint understanding of the combined problem/solution domain.

Developers utilise rapid prototyping and exploratory design approaches to assist clients in initially understanding the problem domain, and how this relates to potential solutions. The result is a specification that, in the first stage of development, evolves to progressively incorporate both requirements and design elements. The in the second stage of development the specification again evolves, but this time in a more conventional iterative development ([Dart, 2000](#)).

As mentioned previously, our earlier work has developed an understanding of the elements that are typically incorporated into a specification, but this has yet to be formulated into an effective model that can be used to underpin the development - both in terms of forming the basis of aspects such as guiding the developing process and supporting specification evaluations, and in terms of underpinning the design of Web development CASE tools.

In the following section we discuss a model of the evolving characteristics of a Web system, and consider how this might be used to support the more effective development of Web systems.

## Characterisation Model

Given the inherent tight coupling between the system requirements and the high level design that results from the exploratory approach to development, and then the evolutionary build cycle, it does not make sense to adopt separate requirements, architecture, and detailed design models - even if these are well linked together. Instead, we are proposing a single model of the evolving characteristics of a system.

We can begin by considering the key elements that such a model should be able to represent.

- The specific characteristics of the system. Particularly we include those elements that, if changed, would impact on the ability of the solution to address the needs of the users and clients of the system.
- Categorisations of the characteristics in a way that facilitates improved understanding of the system and/or enhanced development practices.
- Grouping of the characteristics consistent with required or desired development artifacts.
- Relationships between the characteristics that capture developmental or knowledge dependencies.

We have developed a initial simplified representation of a characteristic model that addresses these requirements. This model can be visualised in various ways. One of the simplest views is shown in Figure 2. This diagram captures many of core characteristics (these are discussed further in the next section - though it is expected that the actual characteristics will change and evolve based on ongoing analysis of commercial practices, changes in technology and infrastructure, and evolution in the broad types of systems that are being developed).

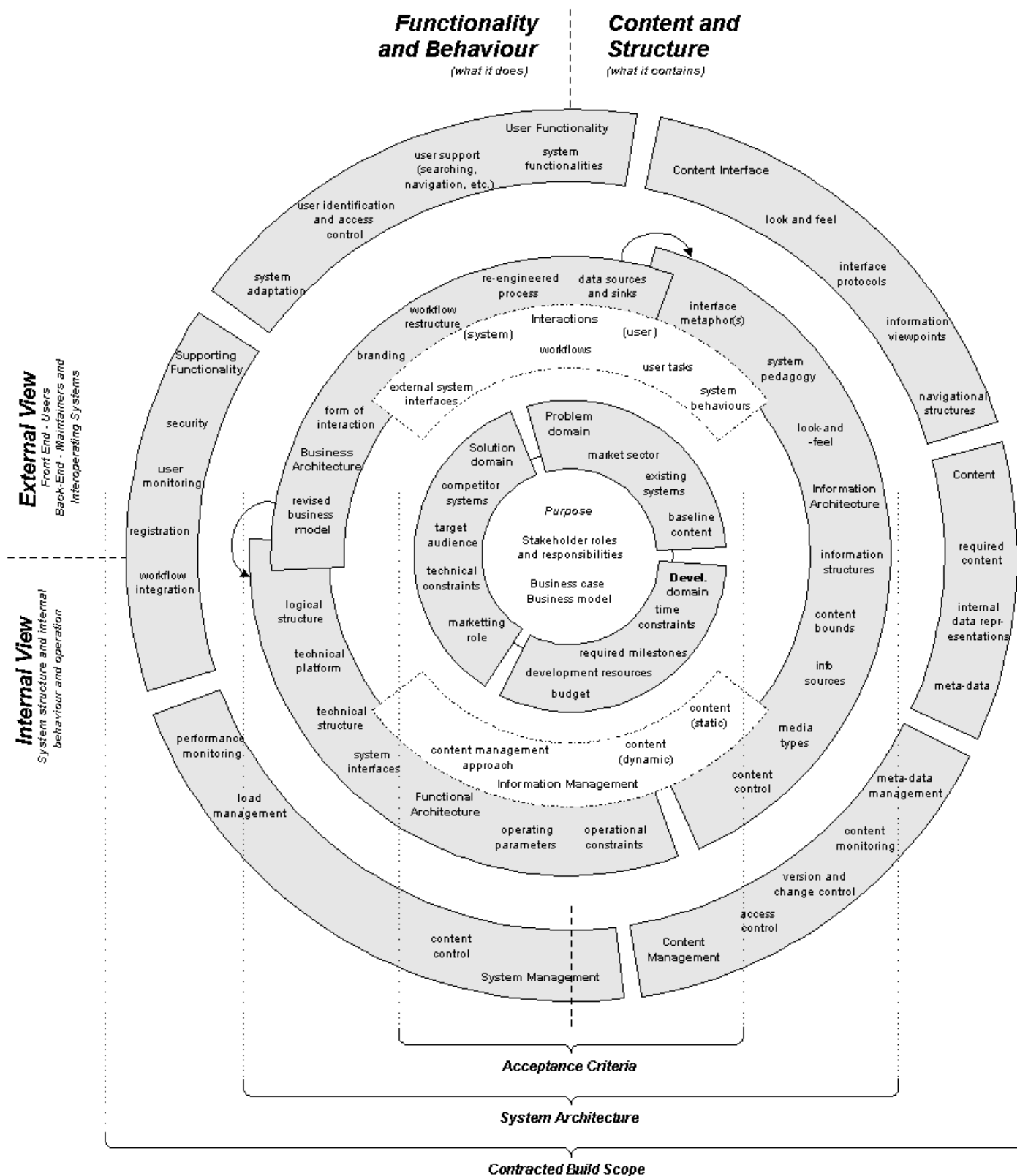


Figure 2: Web system characterisation model

Figure 2 also highlights some of the groupings of characteristics into broad categorisations. It is worth noting that a given characteristic can be categorised in various ways. For example, the characteristic site branding is categorised as part of the business architecture, but can also be viewed as an external characteristic and as a characteristic related to the functionality and behaviour. (It should be noted though, that any visualisation such as this will be constrained in terms of the ability to represent the complexity of the underlying model).

Finally, figure 2 also demonstrates that we can group characteristics into specific target deliverables (or, more correctly, stages in the evolving specification). In particular, the three concentric rings represent the initial acceptance criteria that forms the basis of the initial contract negotiation, the architectural specification that results from the first stage of exploratory prototyping, and the build specification that

results from the second stage of incremental and iterative development. One element not captured in this diagram is the inter-relationships between the characteristics, though the relative positioning of the characteristics does indicate broad dependencies.

We are currently developing a formal representation (as a set of XML schemas) of the model. This will allow it to be more rigorously reasoned about, and (more importantly) used to underpin development tools.

## Specific Characteristics

In order to further refine the model, and to populate it with detailed characteristics, we analysed a number of commercial specifications that were collected from companies that had participated in a series of interviews on Web development practices (mentioned earlier). Specifically, the companies that indicated a high project success rate, and effective client interactions, were approached regarding obtaining samples of specifications for analysis. These companies covered multimedia development, Internet development, and intranet development. Their core businesses varied considerably from consulting and contract development to in-house development. The industries that they serviced also covered a broad spectrum: financial institutions, medical agencies, travel and tourism, legal, manufacturing, government, e-tailers, and consumer advisory services.

Of the companies that identified as internet developers the areas of work ranged from graphic design solutions, e-commerce and database integration to end-to-end solutions. 37% are small companies, 32% medium companies, 26% large national and multinationals and 5% government departments. The smaller companies had an average of 2-4 permanent staff with up to 10 part-time or contractual positions. The medium offices had up to 25 employees and the large offices had up to 145 staff in Australia (and typically 1000's worldwide and with a substantial percentage of those staff working in research and development).

The projects of individual companies varied greatly in complexity and cost (from \$2,000 to \$50million), a reflection of the range of scale of applications being developed for the Web. Predominantly the budgets ranged from \$100,000 to \$1 million large-scale projects, while duration also varied considerably. The average expected lifetime of systems was marginally under 18 months, with few projects expected to have a lifetime of over 24 months.

It is worth noting that most of the respondents attempted to capture requirements before they sign final contracts. It was also recognised that initial tendering often occurs before this point - leading to a two-step specification and contract negotiation (as shown in Figure 1, and discussed in more detail earlier in the paper). Some clients were seen to be happy to commit to a budget (during the tender process) based on broad business objectives, and then finalise the contract at a later stage based on specific analysis of the detailed requirements. The scope of the contracted requirements is typically constrained by retaining a focus on the business case and establishing that there is good basis for specific detailed requirements.

All the respondents had an initial signoff on either a brief or proposal. Only one respondent pointed out that an initial signoff was difficult when they were working off a concept. Essentially, all respondents claimed to have set methodologies for developing complete specifications according to the client expectations, though there was considerable variability in the extent to which these methods were documented.

Based on an analysis of commercial specifications we can make a series of observations, particularly about what is typically specified. Before we look at this however it is also worth noting that the three forms of the specification (acceptance criteria, architectural specification, and build specification - or their variants) were not evident in the artefacts produced by all developers. The broad flow however was consistently based on starting from a set of acceptance criteria (that outlined the business needs, stakeholders and domain characteristics) and then utilising potential designs to develop a clearer understanding of the system. Once this was achieved the actual system build could commence.

So, what characteristics actually did emerge in the development? There was surprising consistency in the views held about the elements that should be captured in initial specifications (i.e. the acceptance criteria). Typical statements from the surveys included:

- Current paper system if one exists. Their vision for the project ~ the client champion for the cause. Access to the potential users.
- Business strategy papers ~ functional requirements ~ documentation ~ workshop outputs
- Business Drivers ~ Business Strategy ~ and Business Requirements for the system from areas affected by the system. :  
With the IT group we look for information on Standards ~ and interface requirements.
- Requirements : business case : budget : stakeholders
- Company info ~ what they would like to do ~ what systems are they looking to improve develop.
- How they operate & produce income ~ markets ~ cutomers ~ products/services ~ current use of technology ~ objectives ~ detailed requirements and needs.

To understand this further, the various system characteristics were extracted and correlated to identify the stage at which they typically emerged. Table 1 lists the key characteristics and the point at which the characteristic emerged.

Table 1: *System characteristics*

Characteristics	Stage of Emergence
<b>Project characteristics</b>	
Project brief	Acceptance Criteria
Stakeholders	Acceptance Criteria
Client profile	Acceptance Criteria
Organisational goals/purpose	Acceptance Criteria
Project business case	Acceptance Criteria
Existing systems	Acceptance Criteria
Potential users	Acceptance Criteria
Legislative compliances	Acceptance Criteria
Development time lines / Deliverables / Budget	Acceptance Criteria
Client role in development process	Acceptance Criteria
Evaluation and deployment processes	Acceptance Criteria
Training requirements	Acceptance Criteria
Quality assurance	Architectural Spec
<b>Business modelling</b>	
Existing business environment	Acceptance Criteria
Client business model	Acceptance Criteria
Branding	Architectural Spec
Workflow restructuring	Architectural Spec
Workflow integration	Build Spec
Form of interaction	Architectural Spec
<b>Informational modelling</b>	
Existing sources of information	Acceptance Criteria
Content bounds	Architectural Spec
Proposed information sources	Architectural Spec
Media types	Architectural Spec
Content control mechanisms	Architectural Spec & Build Spec
Interface metaphor / Look and feel	Architectural Spec & Build Spec
System pedagogy	Architectural Spec
Content structuring / Navigation	Architectural Spec & Build Spec
Information viewpoints	Build Spec
Detailed content and internal data representations	Build Spec
Meta-data	Build Spec
<b>Technical modelling</b>	
Technical constraints	Acceptance Criteria
Logical architecture	Architectural Spec
Technical platform	Architectural Spec
System interfaces	Architectural Spec
Operating parameters and constraints	Architectural Spec
System functionalities	Build Spec
User support mechanisms	Build Spec
User identification and access	Build Spec
System adaptation	Build Spec
Security	Architectural Spec & Build Spec
User monitoring	Build Spec
Registration processes	Build Spec
Performance monitoring and load management	Build Spec

By far the most significant observation is that many aspects that would conventionally be regarded as "requirements" in the sense that they express characteristics of the system that are desired by the client emerge in commercial specifications after design artifacts have appeared. Indeed many of the requirements are actually expressed as part of a design.

It is also interesting to note that, apart from general performance constraints, broad architectural elements, and decisions about technical platforms, most of the functional characteristics often did not emerge until late in the specification process - being deferred to the specific build specification. Conversely, many of the informational characteristics emerged somewhat earlier in the process. For example, decisions about media types, interface metaphors, look and feel, content structure were all made at the architectural level, before the key system functionalities were finalised. This reflects a perception about the importance of the user experience in determining the ultimate quality of a Web system.



## Conclusions And Further Work

In this paper we have presented a characterisation model that can form the basis of effective modelling of Web systems, taking into account the differences between Web development processes and those of more conventional IT systems. In particular, it address the evolutionary and exploratory nature of Web development.

Ongoing work is focussing on several areas. The first is evaluating the applicability of the model to supporting the development of evolution of Web specifications. This is being carried out through a prototype instrument. Further work is also looking at developing a formal XML-based definition of the model, and investigating how this can be applied in the context of Web development tools (a CAWE - Computer Assisted Web Engineering - tool rather than a CASE tool?).

The overall aim is to provide enhanced support for a Web development process that acknowledges the evolutionary nature of Web projects, coupled with a high degree of uncertainty by both developers and clients with regard to not only the form that solutions take, but also the exact nature of the problems being addressed.

## Acknowledgements

The author wishes to acknowledge the collaborative funding support from the Australian Research Council, Access Online Pty Ltd and Allette Systems Ltd. Under grant no. C4991-7612. In particular the author is grateful to John Eklund, Vassiliki Elliott, Ross Jeffery, Louise Scott, Lucila Carvalho, John D'Ambra, Nick Carr and Marcus Carr for their contributions to this research project.

## References

- Angelique, E. (1999, October 24-30). "A Lightweight Development Process for Implementing Business Functions on the Web". Procs. *WebNet'99*, Honolulu, Hawaii, USA, pp262-267
- Baumeister, H., Koch, N., & Mandel, L. (1999). "Towards a UML Extension for Hypermedia Design". Procs. < > 1999: *The Second International Conference on The Unified Modeling Language*, Fort Collins, Colorado, USA, pp614-629
- Beck, K. (1999). *Extreme Programming Explained*, Addison-Wesley.
- Burdman, J. (1999). *Collaborative Web Development*, Addison-Wesley.
- Ceri, S., Fraternali, P., & Bongio, A. (2000, May). "Web Modeling Language (WebML): a modeling language for designing Web sites". Procs. *WWW9 Conference*, Amsterdam.
- Conallen, J. (1999). *Building Web Applications with UML*, Addison-Wesley.
- Dart, S. (2000). *Configuration Management: The Missing Link in Web Engineering*, Artech House.
- De Troyer, O., & Leune, C. (1997, April). "WSDM: A user-centered design method for Web sites". Procs. *7th International World Wide Web Conference*, Brisbane, Aust, pp85-94
- Erskine, L., Carter-Tod, D., & Burton, J. (1997). "Dialogical techniques for the design of web sites", *International Journal of Human-Computer Studies*, 47, pp169-195.
- Fournier, R. (1999). *Methodology for Client/Server and Web Application Development*, Yourdon Press.
- Gates, L. (2001). "Analysis and Design: Critical yet Complicated", *Application Development Trends*, February, pp40-42.
- Gellersen, H.-W., Wicke, R., & Gaedke, M. (1997). "WebComposition: An Object-Oriented Support System for the Web Engineering Life Cycle", *Computer Networks and ISDN Systems*, 29, pp8-13.
- Graham, I., Henderson-Sellers, B., & Younessi, H. (1997). *The OPEN Process Specification*, Harlow, U.K., Addison-Wesley.
- Haggard, M. (1998). *Survival Guide to Web Site Development*, Microsoft Press.
- Haire, B., Henderson-Sellers, B., & Lowe, D. (2001, 8-12 Oct). "Supporting web development in the OPEN process: additional tasks", Procs. *COMPSAC'2001: International Computer Software and Applications Conference*, Chicago, USA.
- Henderson-Sellers, B., Haire, B., & Lowe, D. (2001). "Adding Web Support to OPEN", *Journal of Object*

Lord, J. (2000, June). *Patterns for e-business: Lessons learned from building successful e-business applications*, IBM White Paper. [\[HREF4\]](#)

Lowe, D., & Eklund, J. (2001a, 22-23 November). "Commercial issues in specification of Web systems", Procs. *AWRE'2001: 6th Australian Workshop on Requirements Engineering*, Sydney, Australia, pp4-13

Lowe, D., & Elliott, V. V. (Submitted). "Web Requirements: An Overview", *Requirements Engineering Journal*.

Lowe, D., & Henderson-Sellers, B. (2001b, 6-12 Aug). "Impacts on the development process of differences between web systems and conventional software systems", Procs. *SSGRR 2001: International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, pp21

Lowe, D., & Henderson-Sellers, B. (2001c). "Web Development: Addressing Process Differences", *Cutter IT Journal*.

Overmyer, S. (2000). "What's Different about Requirements Engineering for Web Sites?", *Requirements Engineering Journal*, 5(1), pp62-65.

Russell, P. (2000, November 20-23). "Infrastructure - Make or Break your E-Business", Procs. *TOOLS-Pacific 2000: Technology of Object-Oriented Languages and Systems*, Sydney, Australia.

Schwabe, D., & Rossi, G. (1995). "The Object-Oriented Hypermedia Design Model", *Communications of the ACM*, 38(8), pp45-46.

Takahashi, K., & Liang, E. (1997). "Analysis and Design of Web-based Information Systems", Procs. *7th International World Wide Web Conference*, Brisbane, Aust.

Thomas, D. (2000, June). "Managing Software Development in Web Time Software", Procs. *XP2000*, Cagliari, Italy.

## **Hypertext References**

HREF1

<http://www.eng.uts.edu.au/~dbl/>

HREF2

<http://www.eng.uts.edu.au/>

HREF3

<http://www.uts.edu.au>

HREF4

<http://www-4.ibm.com/software/developer/library/lessons/>

## **Copyright**

David Lowe, © 2002. The authors assign to Southern Cross University and other educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to Southern Cross University to publish this document in full on the World Wide Web and on CD-ROM and in printed form with the conference papers and for the document to be published on mirrors on the World Wide Web.