Linking Modelling Capabilities and Abstraction Levels: The Key to Web System Architectural Integrity

Alice Gu, David Lowe, Brian Henderson-Sellers

University of Technology, Sydney
PO Box 123 Broadway
NSW 2007 Australia

agu@westpac.com.au, david.lowe@uts.edu.au, brian@it.uts.edu.au

ABSTRACT

In response to the unique characteristics of Web systems, suitable software engineering approaches need to be utilized to achieve the quality attributes that are specifically required by Web system development. As an important part of this supporting environment, modelling languages for Web systems need to support not only the modelling of both information architectures and functional architectures but also, more importantly, the integration of these in a cohesive and consistent manner. The key to achieving this architectural integrity of Web systems is the proper linkage between these two major modelling capabilities (for information and functionality) and the various abstraction levels which necessarily occur in such systems.

KEYWORDS

Web modelling languages, modelling capability, abstraction level, architectural integrity, technical architecture, information architecture, functional architecture.

1. INTRODUCTION

The development of Web systems needs to include consideration of two critical aspects of technical architecture: the information architecture and the functional architecture, which both cut across several standard architectural elements and can be captured at various abstraction levels.

A key element in the ability to support the development of effective Web systems is the availability of an appropriate Web modelling language (WML)[1]. Such a WML needs to support the modelling of both the information architecture and the functional architecture of Web systems at various abstraction levels. Furthermore, to maintain Web system architectural integrity, it needs to provide the capability to link the modelling artefacts at various abstraction levels in a cohesive and consistent fashion (Atkinson & Kuhne, 2001; Lowe and Henderson-Sellers, 2002). According to our study of the most prominent WMLs (Gu, 2001), the support for both information architectures and functional architectures is less than satisfactory. In (Gu *et al.*, 2002), we explored the language issues in terms of a gap analysis between current WMLs and a requirements framework. Here, we focus on a related issue – that of discrepancies, omissions and general lack of support for the proper linkage between abstraction levels when modelling Web system artefacts. By analysing and highlighting this gap in current modelling approaches, we identify an area that is in urgent need of further research. We contend that a better linkage between modelling capabilities and abstraction levels in Web system architecture design can ultimately lead to the improvement in Web system architectural integrity, which forms the foundation to many other quality attributes.

[1] We acknowledge that the acronym WML is used in other situations: in the XML community for Wireless Markup Language and as the name for a freely distributed web tool for HTML generation. Here we use it simply as an abbreviation for web modelling language.

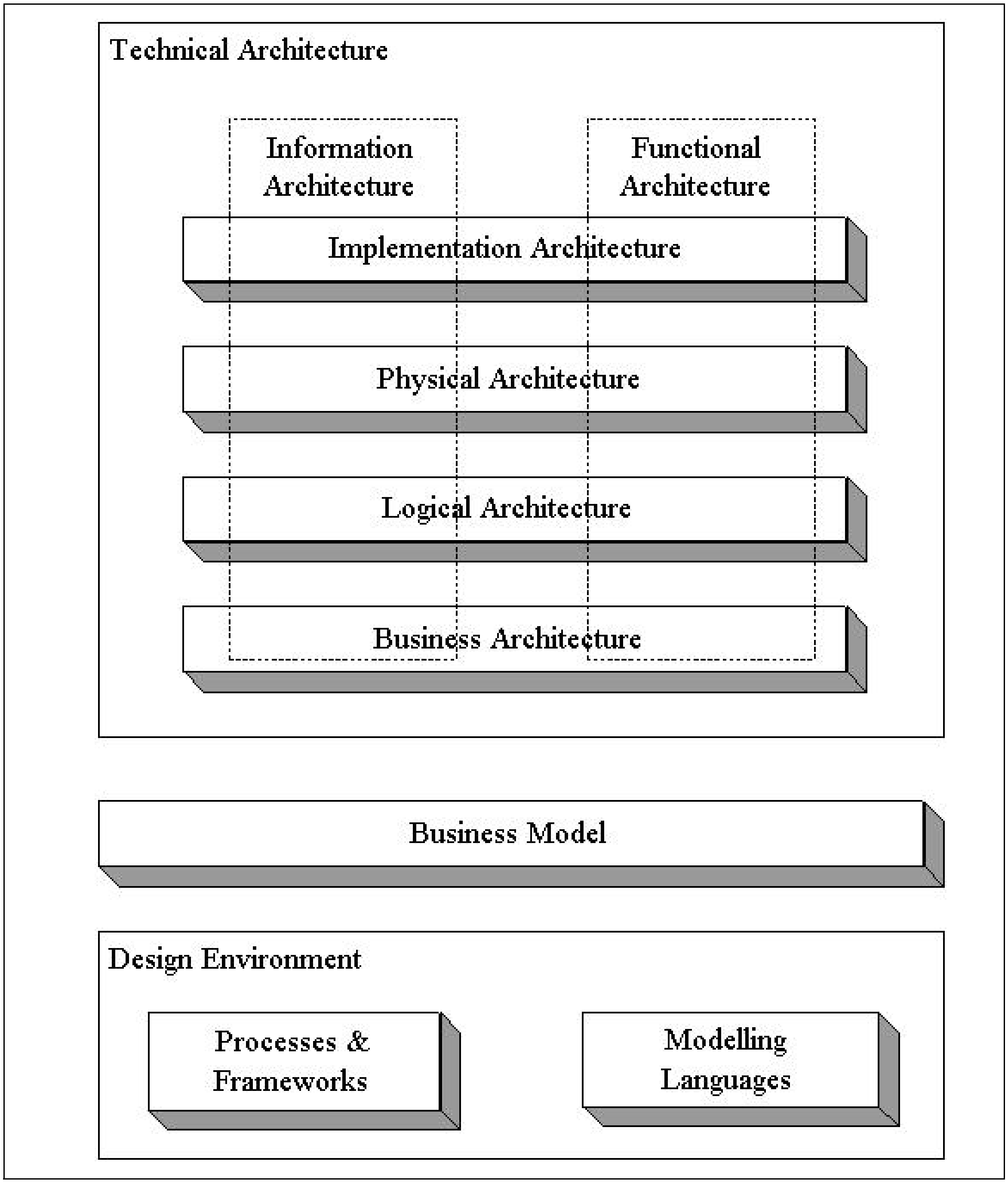

Figure 1 Architectural Design Framework for Web Applications

2. ARCHITECTURAL DESIGN FRAMEWORK FOR WEB APPLICATIONS

A good model should be able to capture, represent and communicate the system architecture at various abstraction levels. Modelling artefacts that are at a high abstraction level (i.e. coarse-grained) are usually used to facilitate the understanding, design and communication of high-level system structure, whereas the modelling of artefacts at a highly detailed level can provide the capability to study various aspects of the system architecture more thoroughly and unambiguously.

To address issues that are specific to Web systems and to support Web system development, modelling languages need to provide the capability to model system architectures at various abstraction levels. Furthermore, modelling artefacts at different abstraction levels should not exist in isolation. They need, instead, to form an integrated and consistent model that addresses Web system development and maintenance issues at suitable levels of abstractions.

As shown in Figure 1, we defined one possible structure for these abstraction levels (detailed below) within the technical architecture, along with their relationship to the two major modelling capabilities of Web system architectures (functionality and information). Figure 1 also shows that these aspects are all managed in the context of both a business model that defines the fundamental business strategy and operational model, and a design environment that incorporates both modelling languages, and processes and frameworks (Gu, 2001). We contend that the design of Web systems should focus on a coordinated and structured combination of the information architecture and the functional architecture. For both the information architecture and the functional architecture, various abstraction levels exist, which include the business architecture, logical architecture, physical architecture and implementation architecture.

- *Business Architecture*: The business architecture represents business processes, policies and procedures, workflows and user interactions. This is used to guide the design of other more technical related architecture layers.
- *Logical Architecture*: The logical architecture defines at a high level the structure of the system to be developed. The elements in this layer are logical concepts, instead of concrete or physical software components.
- *Physical Architecture*: The physical architecture further defines the technical solution at a detailed level. Some design decisions, such as the selection of content storage product, can be represented in this layer.
- *Implementation Architecture*: The implementation architecture specifies system composition and interconnections.

The architectural design of Web systems needs to be conducted with the assistance of tools and techniques related to both development processes and modelling languages. They are both parts of the design environment of Web system architectural design and will be discussed in the next section.

3. MODELLING CAPABILITIES VERSUS ABSTRACTION LEVELS

From the review of existing modelling approaches (Gu, 2001), we can divide the approaches into four different categories, each with a different emphasis in terms of functional versus informational, and detailed versus abstract.

- Category 1: Modelling functional architecture at detailed level;
- Category 2: Modelling functional architecture at abstract level;
- Category 3: Modelling information architecture at detailed level; and
- Category 4: Modelling information architecture at abstract level.

Figure 2 shows a graphical representation of these categories and the extent to which current modelling approaches integrate the models across these categories. In particular, we can look at each of the connections between these categories as follows:

- The connection between category 1 and category 2 is supported reasonably well. In other words, the modelling language support for various abstraction levels in system functional architecture is reasonably rich and mature. This is partly due to the heritage from the modelling of functional aspects in conventional software development.
- The connection between category 1 and category 3 needs improvement. Although some approaches support both functional architecture and information architecture at a detailed level, the linkage between these two aspects in the system architecture is not fully addressed from a modelling language perspective. The fact that the modelling of Web system information architectures and of functional architectures was generally developed from different origins may have contributed to this lack of connection.
- The connection between category 3 and category 4 is absent. According to our analysis, although the support for information architecture is reasonably rich in some approaches that come from a hypermedia background (such as (Ceri *et al.*, 2000) and (Schwabe and Rossi, 1998)), they tend to address design issues at a detailed level. As a consequence,

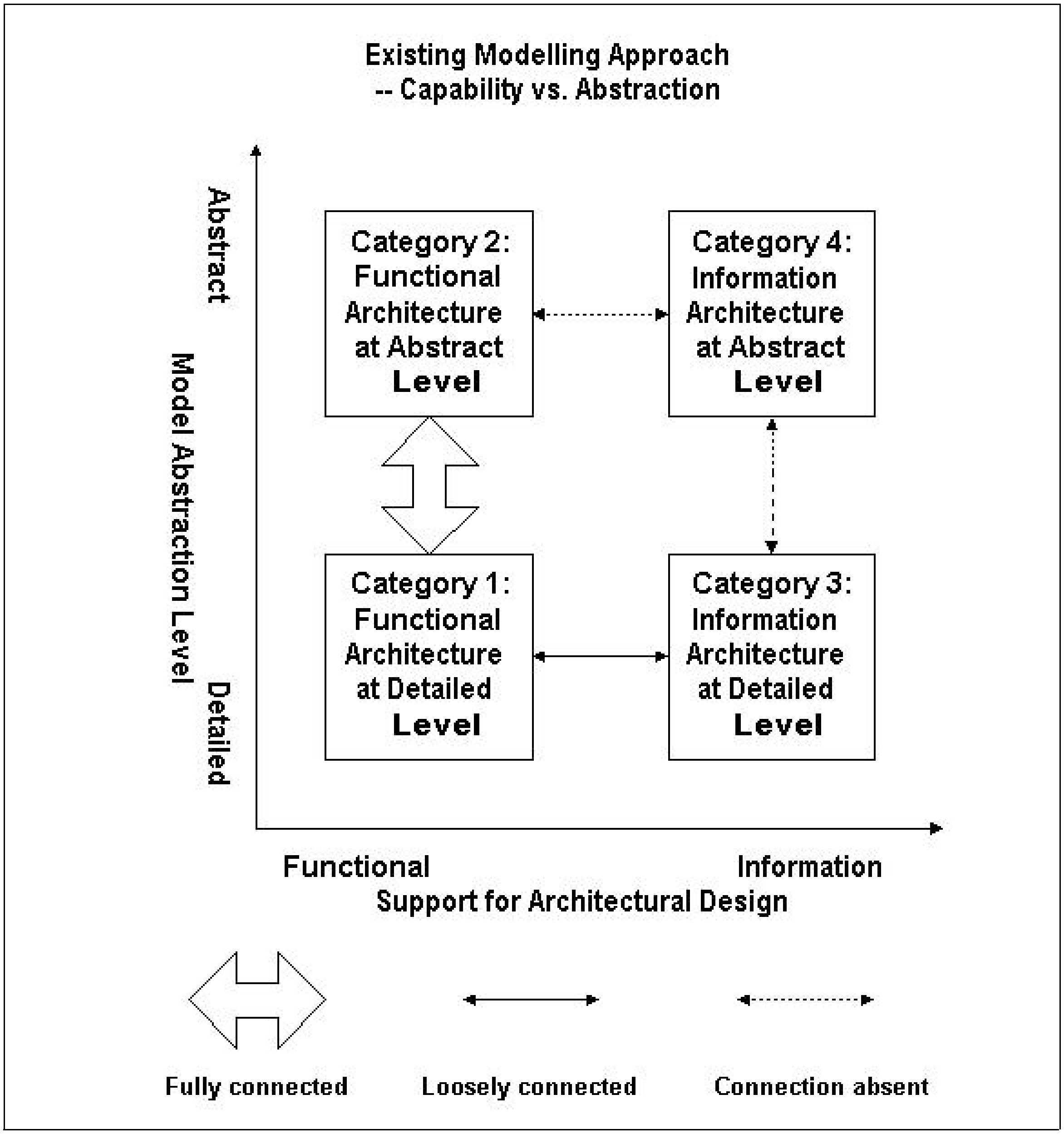
Existing Modelling Approach
-- Capability vs. Abstraction
Abstract
Model Abstraction Level
Detailed
Category 2: Functional Architecture at Abstract Level
Category 4: Information Architecture at Abstract Level
Category 1: Functional Architecture at Detailed Level
Category 3: Information Architecture at Detailed Level
Functional
Information
Support for Architectural Design
Fully connected
Loosely connected
Connection absent

Figure 2 Existing Modelling Approach – Capabilities versus Abstraction Levels

the support for information architecture at a highly abstract level, and thus the linkage between abstraction levels inside information architecture, is still to be properly addressed.

- The connection between category 2 and category 4 is absent. This is partly due to the lack of support for modelling artefacts at an abstract level in the information architecture, and partly due to the lack of understanding of, and in turn support for, the interrelationship between information architecture and functional architecture at a high level during system design.

4. CONCLUSIONS

In this paper, we have considered the current status of Web modelling languages, and particularly the extent to which they integrate modelling of functional and information aspects, and the extent to which various levels of abstraction are considered and connected. Overall, the support within specific categories (such as detailed information models) has evolved to an acceptable level, but the integration between categories is still lacking. We contend that work needs to be conducted in addressing these identified issues to improve Web system architectural integrity and, in turn, other desirable quality attributes.

5. ACKNOWLEDGMENTS

This is Contribution number 02/11 of the Centre for Object Technology Applications and Research at the University of Technology, Sydney.

6. REFERENCES

Atkinson, C. and Kuhne, T. (2001), The essence of multilevel metamodelling, in «UML»2001 – The Unified Modeling Language. Modeling Languages, Concepts, and Tools (eds. M. Gogolla and C. Kobryn), LNCS 2185, Springer-Verlag, Berlin, 19-33

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

Gu, A. (2001), Extending object-oriented modelling languages for web applications, MSc Thesis, University of Technology, Sydney, 182pp

Gu, A., Henderson-Sellers, B. and Lowe, D. (2002), Web modelling languages: the gap between requirements and current exemplars, submitted for publication

Lowe, D., & Henderson-Sellers, B. (2002). Characterising Web Systems: Merging Information and Functional Architectures. In S. Nansi (Ed.), *Architectural Issues of Web-Enabled Electronic Business*, Idea Group Publishing

Schwabe, D. and Rossi, G. (1998), Developing Hypermedia Applications using OOHDM. Workshop on Hypermedia Development Processes, Methods and Models (Hypertext'98), Pittsburgh, USA.