Chapter VIII
Evolving Web Application Architectures: From Model 2 to Web 2

David Parsons
Massey University, New Zealand

ABSTRACT

This chapter explores how Web application software architecture has evolved from the simple beginnings of static content, through dynamic content, to adaptive content and the integrated client-server technologies of the Web 2.0. It reviews how various technologies and standards have developed in a repeating cycle of innovation, which tends to fragment the Web environment, followed by standardisation, which enables the wider reach of new technologies. It examines the impact of the Web 2.0, XML, Ajax and mobile Web clients on Web application architectures, and how server side processes can support increasingly rich, diverse and interactive clients. It provides an overview of a server-side Java-based architecture for contemporary Web applications that demonstrates some of the key concepts under discussion. By outlining the various forces that influence architectural decisions, this chapter should help developers to take advantage of the potential of innovative technologies without sacrificing the broad reach of standards based development.

INTRODUCTION

Web applications grew out of the World Wide Web in the 1990s, driven by the need, particularly within e-commerce applications, for dynamic content. Early Web content, limited to static hypertext markup language (HTML) pages, could not support on-line sales, transactions, personalisation, or any of the other features of the Web that we now take for granted. Subsequently, technologies that could build Web pages on the fly, such as common gateway interface (CGI) scripts, active server pages (ASPs) and Java Enterprise Edition components like servlets and JavaServer Pages (JSPs), transformed the landscape of the Web by introducing the Web application, rather than just the Web site.
Evolving Web Application Architectures

The cycle of innovation that saw the change from static to dynamic content in the mid 1990s has continued unabated. The post ‘dot com’ era has seen Web application design move into a number of new areas. On the one hand we see a rapid evolution of the Web client into the mobile, cross-platform space, with mobile browser-hosted technologies such as XHTML-MP (extensible hypertext markup language—mobile profile), the standard mark-up promoted by the Open Mobile Alliance, a consortium of commercial interests in the mobile communications industry, and micro browsers like Opera Mobile that run on top of a Java Micro Edition environment. On the other hand we see a change on the server side from the ‘walled garden’ Web application, providing only content from a single source, to open Service-Oriented Architectures that enable disparate services from many sources to be integrated by exchanging data using the extensible markup language (XML).

On both client and server we see service driven ‘mashups’ (re-used, intermingled services) and the programmable Web, aspects sometimes associated with the umbrella term the Web 2.0. These changes raise questions about the future of the Web application, for example, can we realistically consider both a rich client and a mobile application that will work across many different types of device? How do extensible markup language (XML) based technologies like Web services and XSLT (extensible stylesheet language transformations) fit in to Web application architecture? How will the Web client evolve with the increasing use of browser hosted applications and Ajax (Asynchronous JavaScript and XML)? Much of the discussion around the Web 2.0 focuses on the social networking aspects of Web based applications, but there is an equally important discussion to be had about the underlying architectures and technologies of applications in the Web 2.0 era. In this chapter we try to address some questions about how Web application architecture continues to evolve, what forces come into play, and how apparently conflicting paths of development may converge into new approaches. Illustrative examples are provided using the Java Enterprise Edition and supporting open source tools.

THE WEB FRAGMENTATION CYCLE

One of the characteristics of the evolution of the Web is what we might term the Web fragmentation cycle, which is the effect of technology driven change in two conflicting directions. On the one hand, we have seen that a particular Web technology, for example a particular type of browser or server side application, can drive Web evolution using features specific to that technology. An early example of this was the introduction into Netscape Navigator of HTML tags for presentation (Lie & Bos, 1999). Netscape’s development of LiveScript (later JavaScript), and Microsoft’s introduction of the XMLHttpRequest into Outlook Web Access (Van Eaton, 2005) can also be seen in this light. On the other hand we see proliferation of Web access tools that increase the reach of the Web across many different types of client. One example of this was the consortium of mobile phone network operators that joined in the WAP (wireless access protocol) Forum in the late 1990s to enable Web access via a wide range of mobile devices by introducing the WAP browser and the wireless markup language (WML).

The effect of technology specific innovation is to narrow the accessibility of Web content to those who have the appropriate technology. While these innovations may increase qualities such as usability and functionality for some, they will exclude those who do not have the right technology. In contrast, the effect of wider reach is to encourage generic technologies for Web access that enable more types of client to access Web-based content, while potentially decreasing usability and functionality due to the need to run on a lowest common denominator platform. The resolution to this dichotomy has traditionally been the introduction of standards on a post hoc