

Assessing the Benefits of AJAX in Mobile Learning Systems Design

Feng Xie

Submitted in partial fulfilment of the requirements for a
Master of Information Sciences at Massey University
Auckland, New Zealand

7th December 2007

STATEMENT OF ACADEMIC INTEGRITY

I declare that this research study is entirely the product of my own work and that it has not been taken from the work of others. When the work and ideas of others has been used in the study, the work has been properly cited in the text.

Feng Xie

Low Risk Notification Statement

This project has been evaluated by peer view and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees. The researcher(s) named above are responsible for the ethical conduct of this research.

If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher(s), please contact Professor Sylvia Rumball, Assistant to the Vice-Chancellor (Ethics & Equity), telephone 06 350 5249, e-mail Humanethics@massey.ac.nz.

ABSTRACT

Today, mobile technology is rapidly changing our life with increasing numbers of services supported by mobile phones, including mobile Internet access and Web-based mobile learning. The growth of the wireless Internet technology opens new path for people to study in anytime and any location. Using Web-based mobile application to present learning resources for mobile learners is a challenge for developers, because the mobile Internet access performance over GPRS networks is often unacceptably slow. A new Web development model, Ajax, may help to address this problem. Ajax (Asynchronous JavaScript and XML), is a new desktop approach to Web application development that uses client-side scripting to provide a seamless user application experience and reduce traffic between client and server. In this paper, we address the question of whether mobile Ajax provides measurable performance advantages over non-Ajax mobile learning applications. A real-life Web-based mobile learning application performance over a GPRS network study was done based on comparing an Ajax application and an Active Server Pages (ASP) application with identical functionality. Our results suggest that mobile Ajax can reduce the bandwidth requirement by 71%, and cut the server's response time in half. In addition, these performance improvements were noticed by users in our small group usability test.

ACKNOWLEDGEMENTS

Sincere thanks to the Institute of Information and Mathematical Sciences, Massey University, for their generosity in letting us use their computing resources to conduct our experiments.

Table of Contents

ABSTRACT.....	3
ACKNOWLEDGEMENTS.....	4
CHAPTER 1: INTRODUCTION.....	8
1.1 Introduction.....	8
1.2 Purpose and potential contributions.....	12
1.3 Organization of the report.....	14
CHAPTER 2: BACKGROUND.....	16
2.1 Internet Technologies and Protocols.....	16
2.2 Mobile Technologies and Protocols.....	19
2.3 Traditional Web client-server communication model.....	22
2.4 Asynchronous JavaScript and XML (Ajax).....	25
2.5 Ajax Engine.....	32
2.6 Mobile environment.....	35
2.6.1 Mobile devices.....	35
2.6.2 Mobile technology.....	37
2.6.3 Mobile Web browser.....	39
2.7 Alternative approaches to Ajax.....	44
2.7.1 Java/XML approach.....	44
2.7.2 Macromedia’s Flash and ActionScript approach.....	45
CHAPTER 3: LITERATURE REVIEW.....	46
3.1 Mobile learning.....	46
3.2 Ajax Implementation.....	49
3.3 Ajax performance on the desktop.....	51
3.4 WWW performance over GPRS network.....	53
CHAPTER 4: METHODOLOGY.....	55
4.1 What is the design research method?.....	55

4.2 Why use the design research method?	56
4.3 Apply design research method process in this research.....	58
4.4 User evaluation design.....	62
4.4.1 Within-subject design	62
4.4.2 Observational Techniques.....	63
CHAPTER 5: IMPLEMENTATION	66
5.1 The Mobile learning application.....	67
5.1.1 ASP mobile learning system.....	68
5.1.2 The Ajax mobile learning system	69
5.1.3 Using the mobile learning system.....	77
5.2 Theoretical performance	79
5.2.1 Theoretical ASP system performance.....	79
5.2.2 Theoretical Ajax system performance	81
5.3 Real-life performance (Vodafone New Zealand GPRS)	84
5.3.1 Vodafone New Zealand GPRS network	84
5.3.2 Real-life performance — Response Size	87
5.3.3 Real-life performance — Response time	90
5.3.4 Real-life performance — Query size	92
5.4 Usability evaluation	96
5.4.1 Usability test overview	96
5.4.2 Usability test	97
CHAPTER 6: Results and discussion.....	100
6.1 Data analysis	100
6.2 Usability test results.....	102
CHAPTER 7: Conclusion	105
7.1 Summary conclusion.....	105
7.2 Recommendations.....	106
7.3 Limitations of the study	106
7.4 Suggestions for future research.....	107
References.....	108
Appendix 1 — Bibliography.....	118

Appendix 2 — Glossary of Term & Acronyms..... 120

Appendix 3 — Usability Testing (Invitation Letter) 122

Appendix 4 — Usability Testing (Checklist) 123

Appendix 5 — Usability Testing (Questions) 124

Appendix 6 — Participants Profile 126

Appendix 7 — All Multi-Choice Questions 128

CHAPTER 1: INTRODUCTION

Ancient saying: "Experience is the father of wisdom and memory the mother."

In this chapter, we will briefly introduce this research, the reason for developing Web-based mobile learning application, the potential contributions, and the organization of this thesis.

1.1 Introduction

Today, development of high techniques represented by information techniques rapidly changes our society from global economy to the personal life. The only way to help you keep up with development is learning. Learning means to gain knowledge by study, instruction or scholarship. It also means the act of gaining knowledge. It can help you gain new knowledge, update your old skills, and understand some new concepts.

Traditionally, people need to go to school to learn from a teacher. However, now distance education allows students to learn far away from the school. Lots of on-line training and learning Websites are on the Internet for desktop or laptop computer users. People can study at home instead of school, but the size of the personal computer makes it hard to be used in a moveable environment. Mobile units can fit into that gap.

Meanwhile, the mobile revolution is changing our life and facilitates our learning process. Mobile learning system over the wireless network is becoming a popular research topic recently. Mobile Learning is a part of the distance education. People can enhance their learning experience by using their mobile devices anytime, anywhere, e.g. mobile phones, PDAs, Pocket PCs, etc. Mobile Learning or know as M-Learning, is an education software or Website that allows you to access learning from your mobile devices. This

software or Website has lessons or activities that can help you learn new knowledge and enjoy the learning process. Mobile learning means people can use their mobile phone or devices to learn anytime, anywhere, even just a few minutes available (e.g. waiting in a bus-stop, waiting for a friend, or travelling). It requests the mobile learning system must have a fast response speed and good user interface to allow user operate the mobile learning system over their small wireless devices (small screen, keyboard and slow wireless connection rate).

The learning process involves a lot of interaction between the teacher and student; even if you learn by yourself, you still need to have some practice to make sure you fully understand the knowledge. Quiz learning system can help learner to improve their personal knowledge and problem solving ability (Yokomoto, 2000). It forces learner to think and learn deeply for every question, answer them carefully, and review all answers at the end. So, a good quiz learning system needs an understandable user interface, high performance to respond all the users' activates. Some good examples are using the quiz learning in their mobile learning system (Black & Hawkes, 2006; Henning, Gina, ussge, Guido, & ling, 2007; Seong, 2006). These requests can be achieved easily on desktop environment, but it is not easy for the mobile learning system, especial for the Web-based mobile learning system.

Furthermore, over the past few years, an increasing number of people have been using mobile Internet access through the wireless network (Church, Smyth, Cotter, & Bradley, 2007). But, due to the limitations of mobile phone hardware, it is still very problematic. There are a number of reasons, for example: small screens, high network latency, low bandwidth, interface complexity, and so on (Rajiv Chakravorty & Pratt, 2002). These weaknesses slow down the speed of the mobile learning process and the performance of the mobile learning system is not acceptable in some case.

Unfortunately, the slow wireless network is a big problem for mobile learning system. Browsing the Website over GSM (GPRS) is a nightmare, and many experience a very disappointing performance. There are various reasons, for example, high and variable latency, fluctuating bandwidth, occasional link 'blackouts' (R. Chakravorty, Cartwright, & Pratt, 2002), packet loss, and link outages. Sometimes, some simple requests are experiencing seconds delay (Stuckmann, Ehlers, & Wouters, 2002).

If the GSM/GPRS wireless network has that many troubles, why we are still using it for the mobile learning system? First, one distinct feature of mobile learning over other learning activities is "mobility" (Leung & Chan, 2003), the GPRS mobile phone network is the most commonly used network all over the world, the widest coverage. Using GSM/GPRS network in the mobile learning system can provide mobile learning services anytime and anywhere around the world. Second, there are plenty of GSM/GPRS handsets and devices on the market low prices. Finally, the GSM/GPRS service fee is cheaper than the 3G wireless network.

To address these issues for mobile devices, we may look to recent developments in desktop browser technologies that might be implemented in the mobile environment to improve the Web-based mobile learning application performance.

In February 2005, a director of User Experience Strategy and a founder of Adaptive Path, Jesse James Garrett introduced us to a new Web application approach - Ajax. It is short for "Asynchronous JavaScript and XML" (Garrett, 2005). Ajax was introduced as a desktop model, not a mobile environment solution, but according to its characteristics (small transmits volume and partial Web page update), it is very useful in the mobile environments, especially in high latency network environments (proper Ajax engine design can reduce the transfer times).

The normal Web application allows users to submit a request; the server will process this request and respond it. Then the client browser will refresh the whole Webpage, even if just a little Web content is changed. But in most cases, the new Web page will be very similar to the old one, which means during these transmissions, some of the network resource is used to transfer replicated data. These data had been downloaded once and save by in web browser's cache. Replicate data download wastes a lot of network resources and users' time.

Ajax uses a clever Ajax engine to handle all the data transmission and the Webpage partial update. The Ajax engine only requests new content from the Web server to reduce the unnecessary transmission. It also can partially update the loaded Webpage, whenever the response arrived. Besides, the Ajax engine runs locally on the client side, the Web application response speed is fast and user experience will be improved. On the other hand, Ajax enabled Website can be accessed from the desktop Web browser and mobile Web browser. One Website for two platforms can save huge amounts of resources for the company and improve service for customers.

Today, more and more Websites implement the Ajax approach to their Websites. The pioneer of the Ajax wave is Google, Google Suggest and Google Maps are two good new examples (Figure 1.1); other examples include Google Gmail, New Yahoo Email and so on.

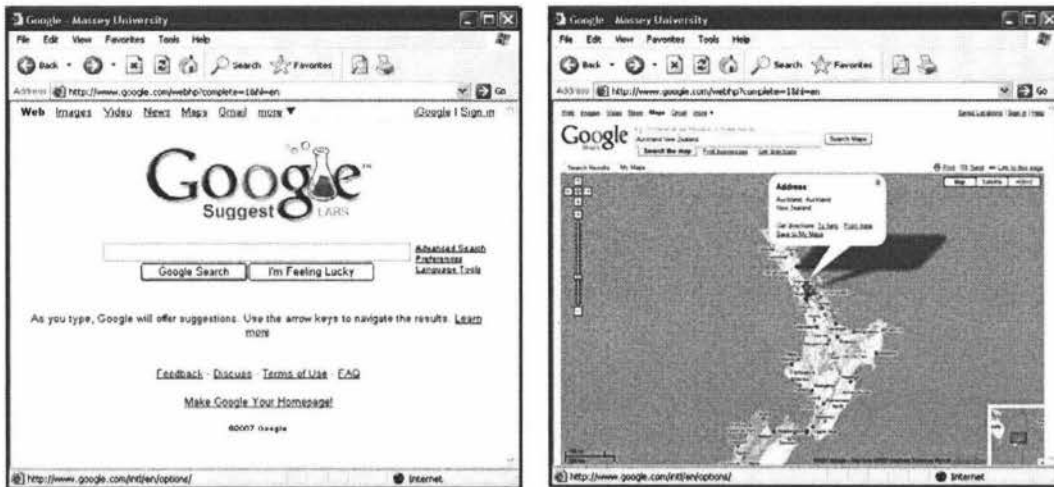


Figure 1.1: The pioneers of the Ajax approach (Google Suggest & Google Maps)

Compared with the traditional Web application model, Ajax can significantly increase a Web page's download speed in the desktop environment. It has been applied by White (2005) in the business realm, and in academia by Smullen and Smullen (2007). Ajax reduces the data transmission volumes between the server and client device, and improves the user experience on the desktop. These are relevant issues for mobile Internet access, which needs a solution to reduce transmission volumes in the context of expensive and low speed connections, and provide a better user experience in browsers with limited screen real estate and navigation tools. Therefore, we might ask the question, if we deploy an Ajax approach to the mobile learning environment, what might be the result? Does Mobile Ajax mobile learning application provide measurable advantages over a non-Ajax mobile learning application?

1.2 Purpose and potential contributions

Over the past decade, moving desktop application towards Web application has gained much attention and acceptance. But today this attention is moving from the desktop Web application toward mobile Web application. We are going to discuss in this thesis

adapting the latest Ajax Web approach to the GPRS mobile environment. We will evaluate the Ajax mobile learning system from performance of system to usability test, and find out what kind of benefits we can gain in this adaptation.

This study will be based on the general methodology of design research. The problem we are facing here is that the traditional Web-based mobile learning system performance over the GPRS network on mobile devices is very slow. So, the alternative suggestion is applying Ajax approach to a mobile quiz learning system into the mobile environment and aim to find out does the Ajax mobile learning system provide measurable advantages over a traditional non-Ajax mobile learning system.

The goal of this research is to applying Ajax to a quiz mobile learning system and evaluates the benefit of Ajax over non-Ajax quiz mobile learning applications, when users access Websites via mobile devices through a General Packet Radio Service (GPRS) network. First, we developed two mobile learning systems with identical functionality, using both an Ajax approach and a non-Ajax approach (using ASP). These two Websites have the same user interface and can be accessed both by desktop and mobile browsers, because they are following the XHTML 1.1 standard. Second, we measured these Websites' performance over a GPRS network based on the data collected from the Web server's log files.

The traditional Website interaction has its limitation, fully page update, long waiting time, logical function only run on the sever side. These weaknesses slow down the speed of the mobile learning process and performance of the Website is not acceptable in some case (Rajiv Chakravorty & Pratt, 2002).

Applying the Ajax approach to a mobile learning system may provide acceptable performance for the users over the GSM/GPRS network. Because, first, the Ajax engine

contents have some logical functions; it can verify the answers locally during self-tests, so it will reduce the amount of transmissions between the Web server and the mobile device, and will also provide a better response speed. Second, the asynchronous and transfer in XML format feature of the Ajax approach allows reduced bytes for every transmission, and provides better response speed. Finally, the Webpage partial update significantly increases the Website response speed.

1.3 Organization of the report

First, in the main part of this report, the Ajax approach will be studied including all related work that has been done recently. Chapter 2 shows the background of the Internet technologies and protocols, traditional desktop Website technologies, and the Ajax approach. For the Ajax approach, it includes the structure of Ajax, the advantages and disadvantages of Ajax and alternative approaches. Also, the background of mobile devices, technologies and Web browser will be introduced in this chapter.

Chapter three outlines all the studies in academic levels related to mobile learning and Ajax, such as the mobile learning, Ajax implementation, Ajax performance on desktop and Web performance over GPRS network. Chapter four will explain the methodology used in the research, the design research method, the reason for using this method, and the process of applying the method in this research.

Second, in chapter five, two mobile learning quiz self-test Web application will be developed. One is Ajax quiz mobile learning system. The other is non-Ajax (ASP) mobile learning system. Chapter five presents the detail of implement the Ajax approach into mobile learning system, the structure of the Ajax Web application, its opponent non-Ajax Web application (ASP Web application). These Web applications can be access from desktop computers and mobile devices.

Later, we measured these two Web applications: Ajax and non-Ajax (ASP) system in the mobile environment. Then, we outlined the evaluation test bed and sub-concluded the experiment results for comparing normal Web application (non-Ajax) and Ajax base Web application based on the server's log. A small group of usability test will be implemented to find out the benefits of Ajax over non-Ajax Websites, when the users access a Web application using a mobile device through General Packet Radio Service (GPRS) network.

In chapter six, a result and discussion for advantages and disadvantage of mobile Ajax Web applications, especially for its performance will present. Finally, chapter seven summarizes the mobile Ajax research, including conclusions, some recommendations, the limitations of this study, and what can we do to upgrade this research into the next level in our future research.

CHAPTER 2: BACKGROUND

Ancient saying: "If you want to understand today, you have to research yesterday".

This chapter outlines all the Internet and mobile technologies related to this research in a time line, and the weaknesses of the traditional Website client-server communication model. Next, the Ajax approach and its advantages will be introduced. This research is applying the Ajax approach into the mobile learning system, so the study of the mobile environment is outlined in this chapter, too. Finally, this chapter outlines the other alternative approaches to Ajax approach.

2.1 Internet Technologies and Protocols

The most popular communication protocol on the World Wide Web (WWW) is Hypertext Transfer Protocol (HTTP). It is used to transfer or convey information over the Internet. HTTP uses a request-response model to communicate between client and server.

Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers (Masinter, 1998). A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred. HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1" (Fielding et al., 1997).

Only having a communication protocol is not good enough for us to surf Internet; we need some tools or languages to talk over the Internet. Hyper Text Markup Language (HTML) is the lingua franca for publishing hypertext on the World Wide Web. It is a non-proprietary format based upon Standard Generalized Markup Language (SGML), and can be created and processed by a wide range of tools, from simple plain text editors - you type it in from scratch- to sophisticated WYSIWYG (What you see is what you get) authoring tools. HTML uses tags such as <h1> and </h1> to structure text into headings, paragraphs, lists, hypertext links, etc (W3C.org, 2007d).

HTML and CSS combined together can provide the user with a stylish and colourful Website. Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colours, and spacing) to Web documents (W3C.org, 2007a). It is a style-sheet language used to describe the presentation of a document written in a markup language, for example, HTML and XHTML. CSS allows users and developers to change the colours, fonts, layout and other aspects of document presentation. The style sheets file is separate from the document content (HTML or XHTML file) and can provide more flexibility and controllability for document presentation.

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of HTML documents. The HTML document can be further processed and the results of that processing can be incorporated back into the presented page (W3C.org, 2007b).

"Dynamic HTML" is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allow documents to be animated. The W3C received several submissions from member companies on the way in which the object model of HTML documents should be exposed to scripts. These submissions do not propose any new HTML tags or style sheet technology. The W3C DOM Activity is working hard to make

sure interoperable and scripting-language neutral solutions are agreed upon (W3C.org, 2007b).

HTML can be used to create Websites, CSS to manage the Website style, DOM to give the Website a dynamic way to update the content; combined, you may have a Website solution. But the HTML is not perfect yet; an updated version of HTML was released and recommended by W3C in 2000. The Extensible HyperText Markup Language (XHTML™) is a family of current and future document types and modules that reproduce, subset, and extend HTML, reformulated in XML. XHTML Family document types are all XML-based, and ultimately are designed to work in conjunction with XML-based user agents (W3C.org, 2007d).

XML is short for Extensible Markup Language (XML) and is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere (W3C.org, 2007c). XML is designed to describe data by defining your own tags. It uses a Document Type Definition (DTD) or an XML schema to describe. HTML and XML are different; XML was designed to describe data and to focus on what data is; HTML was designed to display data and to focus on how data looks (W3Cschools.com, 2007). You can consider XML as a data saving format and HTML is used to control how to display the information.

XHTML Mobile Profile is designed for resource-constrained Web clients that do not support the full set of XHTML features, such as mobile phones, PDAs, pagers and set-top boxes. It extends XHTML Basic with modules, elements and attributes to provide a richer authoring language. The XHTML Mobile Profile is a strict subset of XHTML. It extends XHTML Basic to bring enhanced functionality to application authors, including

additional presentation elements and support for internal style sheets (WAP Forum, 2001).

Like a pair of parallel marks, Internet technology and mobile technology did not cross each other until recently.

2.2 Mobile Technologies and Protocols

The most popular mobile phone standard is Global System for Mobile communications (GSM). It is an open, digital cellular technology used for transmitting mobile voice and data services. GSM differs from first generation wireless systems in that it uses digital technology and time division multiple access transmission methods. GSM is a circuit-switched system that divides each 200kHz channel into eight 25kHz time-slots. GSM operates in the 900MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. The 850MHz band is also used for GSM and 3GSM in Australia, Canada and many South American countries. GSM supports data transfer speeds of up to 9.6kbit/s, allowing the transmission of basic data services such as SMS (Short Message Service). Another major benefit is its international roaming capability, allowing users to access the same services when travelling abroad as at home. This gives consumers seamless and same number connectivity in more than 210 countries. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available (GSM Association, 2007).

As the mobile service booms, GSM Association introduced a new non-voice, high-speed data service—GPRS (General Packet Radio Service). It is the world's most ubiquitous non-voice wireless data service, available now with almost every GSM network. GPRS is a connectivity solution based on Internet Protocols that supports a wide range of enterprise and consumer applications. With throughput rates of up to 40kbit/s, users have

a similar access speed to a dial-up modem, but with the convenience of being able to connect from anywhere. GPRS can be utilized for services such as WAP access, SMS and MMS, but also for Internet communication services such as email and Web access (GSM Association, 2003).

WAP 1.0, Wireless application protocol (WAP) version one is an application communication protocol, used to access services and information for handheld devices. And a protocol defined as an XML 1.0 application, designed for mobile browsers. (WAP Forum, 2002a). WAP service initially launched in Europe during 2000 was not a success (Loken, 2000), because it is expensive and slow.

The biggest advantage of WAP protocol stack is that it is designed to minimize the required bandwidth and maximize the number of wireless network types that can deliver WAP content (International Engineering Consortium, 2007). At the beginning WAP was standard for mobile Web surfing, but its disadvantages are now clear (International Engineering Consortium, 2007). First, the content providers need to pay for the standard, and setup a server for quality WAP service. Second, it is hard to maintain; Website developers need to develop a special Website for WAP, even it is has the same information as the normal Website. Now, WAP/WML is slowly fading out of the mobile content picture (Kumar, Parimi, & Agrawal, 2003).

WAP 2.0 is a next-generation of Wireless application protocols. It also optimizes usage of higher bandwidths and packet-based connections of wireless networks worldwide. While utilizing and supporting enhancements in the capabilities of the latest wireless devices and Internet content technologies, WAP 2.0 also provides managed backward compatibility to existing WAP content, applications and services that comply to previous WAP versions over the latest wireless the technologies known as GPRS and 3G (WAP

Forum, 2002b). It adds support for the standard Internet communication protocols such as TCP/IP and HTTP.

WML (Wireless Markup Language) is a standard mark-up language used by Wireless Application Protocol (WAP) for small, narrowband, mobile devices. WML is a markup language based on XML and is intended for use in specifying content and user interface for narrowband devices, including cellular phones and pagers (WAP Forum, 2000). It supports four major functions, text presentations and layout; deck/card organizational metaphor; inter-card navigation and linking; and string parameterization and state management. WML is similar as HTML (Hyper Text Markup Language) provides Web pages for users. The pages in WML are called “decks” and it is constructed as a set of “cards”.

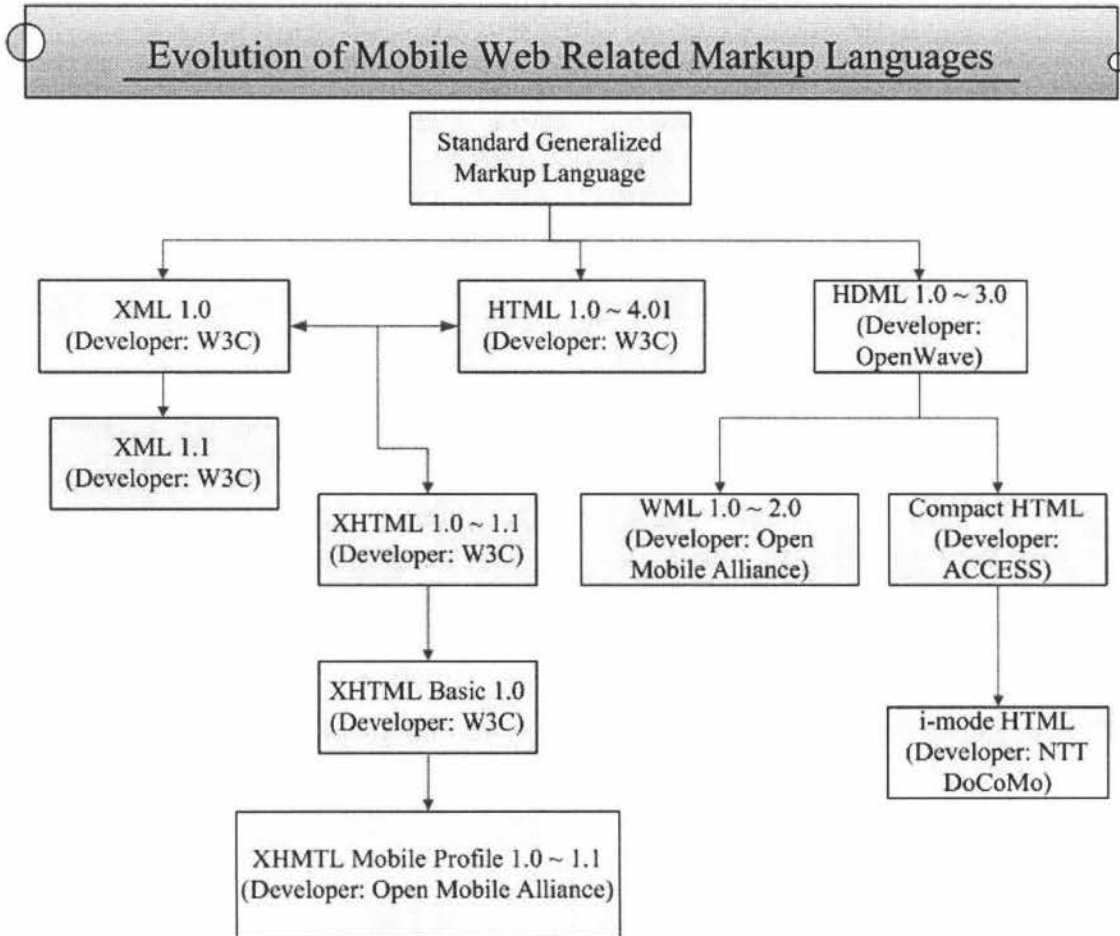


Figure 2.1: Web Mark-up Language Time Line.

This section explains all of the most important technologies and protocols useful for the research from desktop environment to mobile environment. The relationship for the Web Mark-up language can be well explained by Figure 2.1. In the next section, we will briefly explain the disadvantages of the traditional Web client-server communication model.

2.3 Traditional Web client-server communication model

Since the first Webpage built by Tim Berners-Lee in 1992 (T. B. Lee, 1992; Watson, Rainer, & Koh, 1991), “synchronous” cycle reigned in the Website access approach in

the past decade. It represents a “click, wait, and refresh” action approach, or, considered as a communication mechanism, it is a “synchronous request and response” cycle. Users have to follow this rule when they are surfing on the Internet (Wei, 2005). “Click, wait, and refresh” interaction model is a classic and first user interaction model for the Web browser. Like its name, the user needs to click to send a request and wait until the Web page refreshes to a new page.

“Click, Wait, and Refresh” interaction model

Back to the beginning, “click, wait, and refresh” interaction model is the only user-computer interaction model used in the past decade, because the global Internet network is slow. When users surf on the Internet, and he/she want to view a new page, then he/she must do the following action (Figure 2.2).

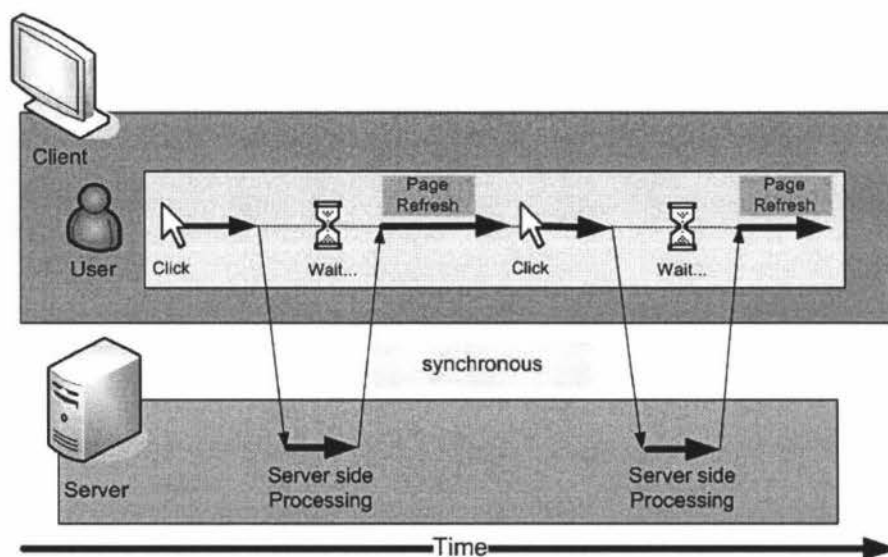


Figure 2.2: Classic Web application model: Full page refresh and synchronous communication. (Wei, 2005)

Click - Click on the hyperlink or submit a form, this request will be sent to the server.

Wait - Then the user starts their wait time. Under some wakeful network

environments, it will be a long time. The waiting time depends on the network environment and the server response speed.

Refresh - After a few seconds or a few minutes waiting, a new HTML page display on the screen.

The user must wait from the time a request to be converted into binary and sent by HTTP over an Internet, processed on the server, received result by HTTP and displayed in the browser(Crane, Pascarello, & James, 2005). Users' waiting time include network transition time and server response time and the browser display time (milliseconds). The problem is users cannot do anything when they are waiting, and this waiting time can be up to minutes depending on the network status. Obviously, this "click, wait, refresh" approach makes a lot of technical sense, it is easy for Web developers to develop the Website, but it is not efficient and not good for users' experience. It forces users to act like a machine, input once, waits for process and output.

"Synchronous request and response" communication model

On the other hand, the traditional Web application communication model is a "synchronous request and response" communication model. While the time is running, the client user initiates a request, and the server responds to it. The communication is always initiated one way from the client to the server. The server never initiates requests. During communication, users cannot do anything; they have to wait until the server responds to the request.

It is like a car running on a freeway connected between the client and the server. The freeway allows two lines of traffic to travel at the same, but due to the approach we are using, at any point in time only one car is allowed to run on the freeway. What a waste!
(Figure 2.3)

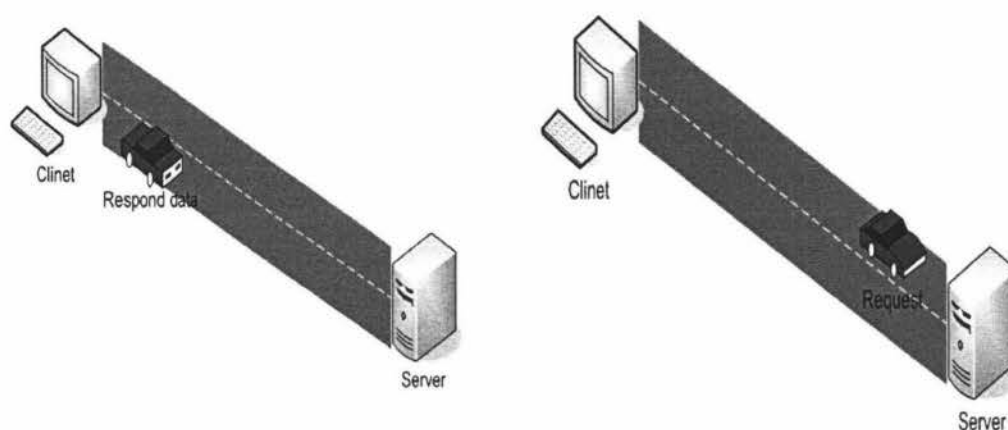


Figure 2.3: Freeway traffic

The traditional Web application cannot provide a seamless user experience. According to Hugh Watson, most users will run out of patience in six seconds due to lack of system performance (Watson, Rainer, & Koh, 1991). Under the limitations of network bandwidth and latency, the only way to improve the Web application performance is to break down the synchronous model, and jump off the “click, wait and refresh” cycle. Then Garrett came up a solution - Ajax.

2.4 Asynchronous JavaScript and XML (Ajax)

What is Ajax? Ajax is short for “Asynchronous JavaScript and XML”. It is a new Web development approach to improve the Website’s performance on the user’s and server’s level, combined with a set of widely-used, powerful and mature Web technologies including JavaScript, XHTML, XML and DOM. Ajax runs in a client-side browser; it is a client-side technology.

Ajax is a style of Web application development that uses a mix of modern Web technologies to provide a more interactive user experience (Sayar, Pierce, & Fox, 2006). For “click, wait, and refresh” and “synchronous request/response” in classic Web applications, it means it is slow, unreliable, inefficient, and has low productivity. In the

Ajax Web application approach, it altered these two basic behaviours and produced a new high performance and more interactive Web application model.

“Partial screen update” takes place over the “click, wait, and refresh” user interaction model. (Wei, 2005)

Traditional Web pages refresh the whole page every time some new information arrives from the server. During this time, users must wait for the page to refresh. Ajax only updates the part of the user interface on the screen that contains new information, retaining the current Web page in the browser.

The “synchronous request/response model” replaced by asynchronous communication model. (Wei, 2005)

The other key feature for Ajax technology is using the new “asynchronous” communication model. Compared with the old “synchronous” communication model (Figure: 2.2), the “asynchronous” communication model allows users to continue to use Web applications without waiting. As a result, requests will continue to be sent out if necessary, but users do not need to wait for the responses. The Ajax engines will automatically request information from the server in the background. Meanwhile, users can keep using other functions of the Web application. When the new information arrives, the Ajax engine will create a HTML Web page to partially update the user interface to respond to users’ requests (Figure 2.4).

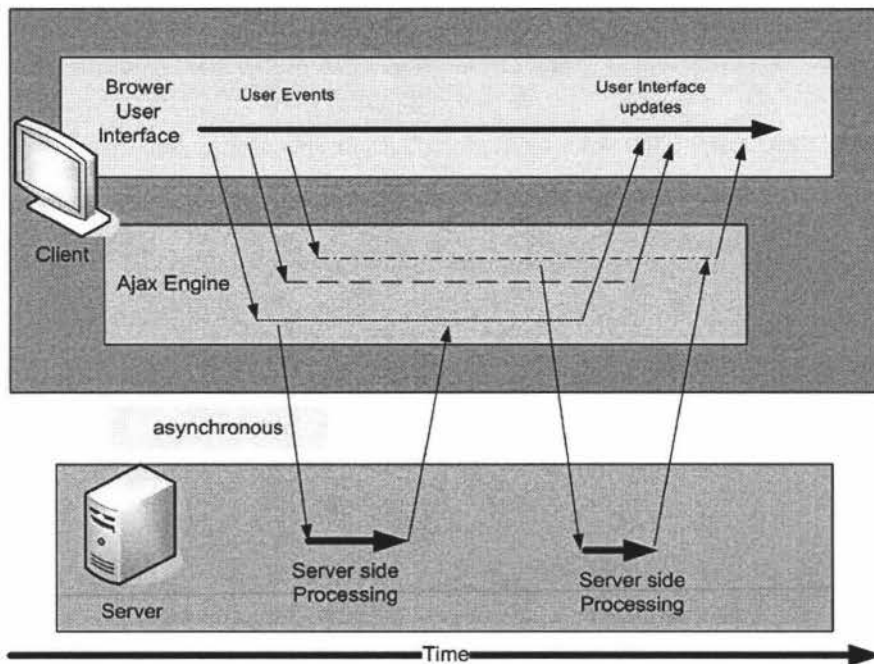


Figure 2.4: Partial user interface updates

Therefore, the user can enjoy a seamless browsing experience. It is like a busy and efficient freeway (Figure 2.5).

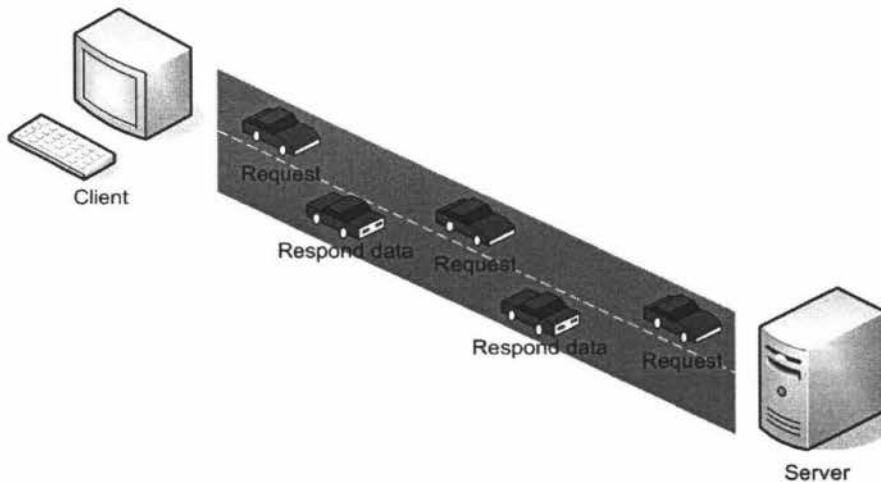


Figure 2.5: Busy freeway traffic

The weaknesses of the traditional Web interaction paradigm and communication model are the strengths of the Ajax model. Ajax Web development approach for the first time

introduces some common features from desktop application to Web-based application. A new door opened for Ajax developers by using “partial update” and “asynchronous” features.

- Instead of submitting the form page by page and submitting it by the user, Ajax does submissions automatically, when users trigger some events.
- Power up client machine’s processing, send out the query asynchronously, then make Web browser receive the results continually.
- Smaller, incremental in-place partial Webpage updates, instead of full-page post back refresh.
- Reduce the size of Web application; Ajax engine only downloads new context.

A common user register system can make these features more clear, for example: A user wants to register an online account; what he/she need to do is open the registration Web page, type in his/her information, such as, account name, password, email address (the system needs to send you a confirmation email), and so on. Whilst typing the information in, the user may not be aware that the account name had been taken, or the password wasn’t secure enough, or the email address he/she is using cannot receive the confirmation email. So, the user still submits uncertain information to the server, and then waits.

When submitted information is accepted by the server, the client’s computer can receive a new “register successful” page. But, if any information is against the server’s applied register policy, the user must redo the registration form. For example, if the user’s username had been taken by someone else, he/she cannot use that user ID, and the rest of the filled in information can be accepted by the server, but the server would understand the difference between one field and all fields’ errors.

Under this scenario, the user needs to refill in the erroneous information and resubmit the information again until the server accepts it; sometimes errors in typing or unclear policies may cause multi-type in and long waiting times. The user wastes time typing in information, and waiting for the server's response.

In a new Ajax register system, the register process will be slightly different. The user opens a registered Web page, just after she types in her preferred account name, the user triggers an event (keydown event); the Ajax engine automatically submits her preferred account name to the server in the background (the user may not even know this). Meanwhile, she can keep typing her password, email address and so on.

When the server receives this automatic request, it will treat it as a normal manual submit request, then verify the account name and respond to this request (invalid or valid account name). When the Ajax engine receives this reply, it partially updates the user interface to notify the user of the verified result. During this transmission, the amount of transmission is very small, only text-base XML (e.g. "valid" or "invalid"; "good" or "bad; "1" or "0") is returned from the server.

Similarly, the same process will be done in password and email address fields. These requests may send out continuously, and Ajax engine will simultaneously partially update the loaded Webpage by response. Users do not need to wait for all the responses to come back; they can still manually submit the form any time they finish.

Therefore, before they actually submit the whole application form, users may notice all the information they need, which fields need to be changed, and what kind of errors they have. As a result, users do not need to follow the "submit and wait" order; they are jumping off the wait queue and experimenting with an exciting user-orientated experience.

When the user has registered successfully, a welcome page will be displayed. Again, in classic HTML style, a whole new “welcome” page is downloaded from the server. The HTML file may include the main menu, policy, font style, and so on. But in the new register system, will be downloaded only a sentence (“Your register was successful. Welcome.”), from the server, and the Ajax engine will update the loaded Webpage to advise the user of the result, and keep the other part of the Webpage.

In summary, asynchronous automatically posting back the request to the server, retrieving data continually and updating a part of the Web page, are key characteristics of Ajax. Theoretically, these features can significantly improve the performance of Web applications and user experience on a desktop platform.

Ajax is a Web development technique for creating interactive Web applications. It is not for a single new technology, it combines a set of powerful, widely-used, and well-known and mature technologies, such as JavaScript, XML, HTTP, and XHTML. Ajax is a client-side browser technology independent of Web server technology or software. The intent of Ajax is to make Web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire Web page does not have to be reloaded each time the user makes small changes. This increases the Web page's interactivity, speed, and usability.

The first article heralds Ajax from Garrett, “Ajax: A new approach to Web applications” (2005). In his essays, he defines Ajax as “Asynchronous JavaScript and XML”; it includes HTML and CSS, Dynamic HTML, XML, JavaScript, asynchronous data retrieval objects (XMLHttpRequest object), combines them together and incorporates them to create a powerful new interaction approach. He uses 5 phrases to explain Ajax (Garrett, 2005):

- Standards-based presentation using XHTML and CSS;
- Dynamic display and interaction using Document Object Model;
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using XMLHttpRequest;
- And JavaScript binding everything together.

The basic Web application interface can be constructed with DHTML or CSS; the document object model (DOM) is used to provide a dynamic Website information solution and interactive user experience; the data format setting between client and server is XML or XSLT (Extensible Stylesheet Language Transformations); XMLHttpRequest object is a key object for Asynchronous client/server communication; and JavaScript acts like a glue to bind all these components together.

These characteristics explain differences between the Ajax Web application and traditional Web application. By using JavaScript to control submission and properties of XMLHttpRequest objects, Ajax allows JavaScript to initiate requests, asynchronous client/server communication, and partial loaded page updates. All functions combined together break down the traditional user interaction “click, wait, refresh” model cycle and reduces users’ waiting time and increases the levels of interaction.

Ajax is a style of Web application development that uses a mix of modern Web technologies to provide a more interactive user experience (Sayar, Pierce, & Fox, 2006). For “click, wait, and refresh” and “synchronous request/response” in classic Web applications, it means it is slow, unreliable, inefficient, and has low productivity. In the Ajax Web application approach, it alters these two basic behaviours to produce a new high performance and more interactive Web application model.

2.5 Ajax Engine

Not only the communication model, but also the structured layers of Ajax are different from the traditional Web client-server communication model (Figure 2.6). Ajax has three structure layers: user interfaces, Ajax engine, and server (Figure 2.7). User interfaces and the Ajax engine run on the client side.

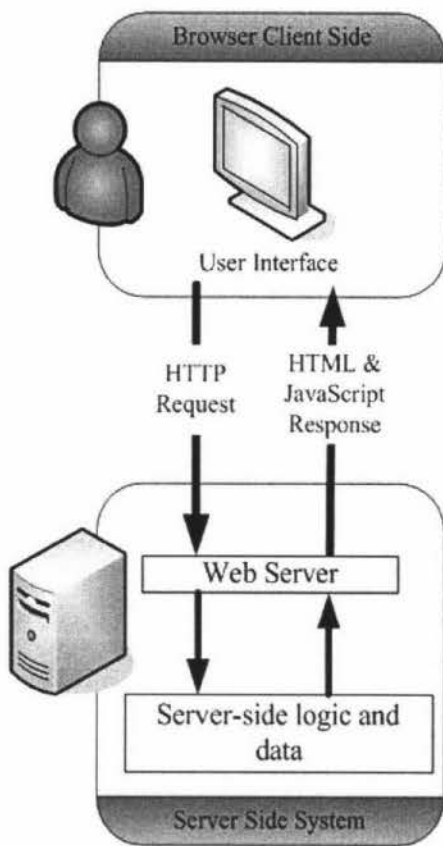


Figure 2.6: Classic Web Application Architecture (Wei, 2005)

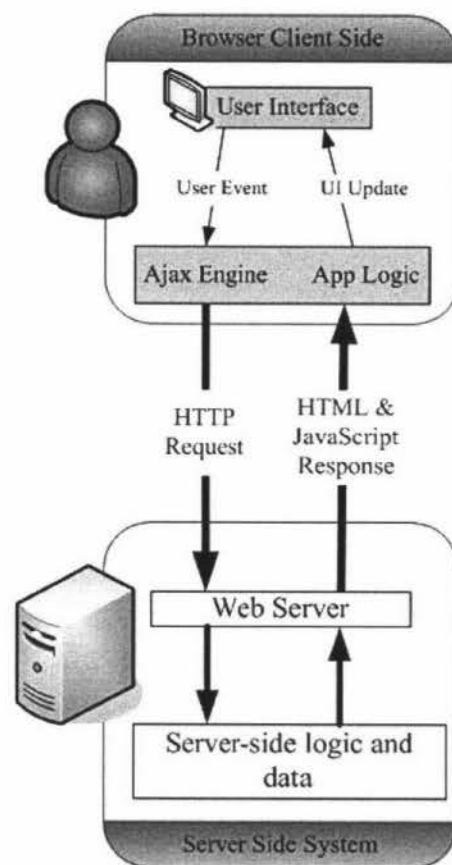


Figure 2.7: Ajax Architecture (Wei, 2005)

Ajax engine as a middleware sitting between the client user interfaces (CUI) and the server. It controls the user interface and the data transmission between client and server, and also some logical functions. When the user initiates his/her request, this request will pass to the Ajax engine first, instead of the server directly in a traditional Client-Server

model. Then the Ajax engine will combine the data from the Web browser cache and the data from the server together to respond to the request. The main advantage of this combining data process is it may reduce the data size loading from the server, and increase the response speed.

Besides, data transfer between the Ajax engine and server is text based XML files, not the whole new Web pages. These text based XML files only contain the content of the Web page; they do not include any Web page structure, colour or layout information. It also can reduce the client-server transfer volume and time.

The key component of the Ajax approach is the Ajax engine. It is created by JavaScript, and saves as a JavaScript file. It usually uses these technologies to control the Web application: JavaScript, Document Object Model (DOM), Cascading Style Sheets (CSS), HTML, and the XMLHttpRequest Object (Figure 2.8). The Ajax engine can be loaded into the client side any time depending on the application designed. Sometimes, it is contained in first Website pages, or when the Web application starts.

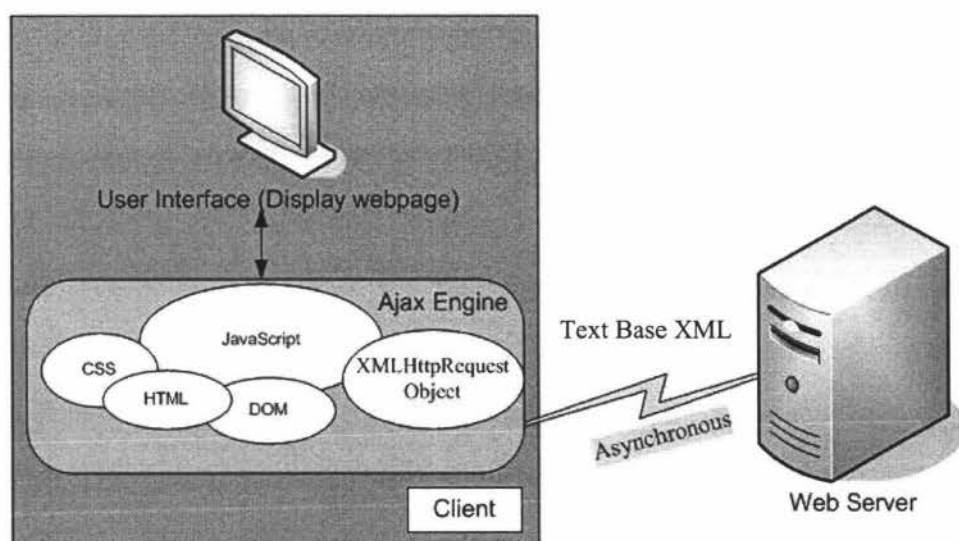


Figure 2.8: Ajax Engine

The XMLHttpRequest Object is an object of the JavaScript. Its specifications are defined an API that implements an interface exposed by a scripting engine that allows JavaScript to perform HTTP client functionality for transferring data between a client and a server, such as submitting form data or loading data from a server (W3C.org, 2007e). By using the XMLHttpRequest object, a Web developer can partially update the Webpage with data from the server after the page has loaded using asynchronous client-server communication.

XMLHttpRequest objects can be used by JavaScript and other Web browser scripting languages to transfer XML and other text data to and from a Web server using HTTP, by establishing an independent communication channel between a Web page's Client-Side and Server-Side. The user's requests are received by the Ajax engine, the engine analyses the request and sends out a new request to the server; when the client sends out the request to the server by XMLHttpRequest object, the server will often fetch data information that came from back-end database and respond in XML data format to the client.

The request from the user interface may ask for a whole new page, but new requests from the Ajax engine only ask for new content; the server returns the content back to the Ajax engine, without any page format or Webpage layout information. Next, the Ajax engine will use these contents to partially update the user interface through HTML, CSS, or DOM.

In traditional HTML Web applications, as mentioned before, all the information is submitted from the user by using the "get" or "post" method. In other words, the "submit" action is manually controlled by the user's "click" action following the "click, wait, and refresh" interaction model.

But with Ajax, the JavaScript can directly communicate with the Web server through the XMLHttpRequest object. It means not only “click events” can activate submit action, but also other particular user’s mouse or keyboard events can activate submit action, such as, onekeyup events, onekeydown events, onechange events, and so on. XMLHttpRequest objects allow asynchronously sent out requests to the server. The requests can be sent out continuously when a user triggers the event, and one does not need to wait until the previous request returns before sending out another request.

The whole picture of the Ajax approach looks like this: an Ajax engine controls the XMLHttpRequest object and is used to asynchronously obtain the latest data or send out the user information between the Web browser and the server. When the new content arrives, JavaScript uses the XMLHttpRequest Object to partially load the Web page using DOM and CSS to supply congruous user interface or Website style.

2.6 Mobile environment

Mobile devices and wireless networks are changing rapidly these days. If applying Ajax to mobile devices, mobile environment research is necessary.

2.6.1 Mobile devices

The key advantages of mobile devices are that they are small, moveable and wireless. In addition, the latest mobile devices have more powerful processors, bigger display screens or multi-screens, colourful user interfaces and integration of multi-communication technology (E.g. Bluetooth, USB, Wi-Fi, etc). The latest mobile devices not only have stylish designs, but also provide us richer user experiences to satisfy our daily lives. Mobile phones and PDAs are the most popular mobile devices today.



Figure 2.9: Mobile phones (Evolution)

Mobile phone

A mobile phone, also known as a cell phone, is a portable mobile device allowing the user to make a call in the cell's signal area, even if the user is moving. The first generation mobile phone used analogue signals; the second generation digital signal is used and more functions were added into mobile phones, for example SMS (Short Message Service) for text messaging between mobile phones, MMS (Multimedia Message Service) for phone and video messaging service (Figure 2.9).

The mobile phone industry is one of the hottest industries in the world today. There are around 2.4 billion mobile phones around the world, and about 800 million sold in 2005; meanwhile, just 200 million PC's were sold in 2005 (BBC.com, 2006), and more than 123 million smart phones were sold in the same year.

Smart Phone and PDAs

Personal digital assistants (PDAs) are referred to as hand-held computers, and also pocket computers. As a computer, it offers a lot of functions that can help you to manage your daily life in digital ways, such as: a calendar your diary; sending and receiving E-mail for your communication; accessing the Internet; recording video; editing documents by PDA

office software; playing computer games; listening to music or radio; and even locating yourself by using the Global Positioning System (GPS) (Figure 2.10).

Latest technologies provide many more functions on PDAs; you can enjoy media, share your files or access the Internet via Wi-Fi technology or Wireless Wide-Area Network (WWAN). When the mobile phone and PDA combine together (smart-phones), you can make a call on your PDA. Most PDAs or smart-phones use touch screen technology to enhance their usability.



Figure 2.10: Personal digital assistants (PDAs) with GPS

The boundaries between mobile phones and smart phones or PDAs have become more and more close. Both of them have cameras, more than 32k colour screens; even touch screens are not rare in the market today; various data connections including GPRS (Brasche & Walke, 1997) and Wireless Fidelity(Wi-Fi). All of these features allow dreams to come true – browse any Website on your mobile phone.

2.6.2 Mobile technology

There are a lot of mobile technologies developed by companies or organizations. It is difficult to say which one is the best or worst, and it is hard to classify them by technology. But, at the most basic level, mobile technologies have gone through four “generations”.

- 1.0G – Analogue. The first generation mobile phone network; the key feature of this generation is it uses analogue signals to transfer the conversation.
- 2.0G – Digital. Using digital signals to transfer the conversation is the most different feature from the first generation.
- 2.5G – It is still considered as second generation, because it still uses a digital signal. Compared with the 2G, 2.5G uses packet switching instead of circuit switching, but the transfer speed is not very high, usually about 20kbps and up to 100kbps, such as, CDMAone, Digital AMPS, GSM/GPRS (Brasche & Walke, 1997).
- 3.0G – The third generation of digital networks can provide a much faster data transfer rate than 2.5G. For example, UMTS can support up to 14.0Mbps in theory; HSDPA provides improved down-link performance of up to 14.4 Mbit/s theoretically.

Today the most common mobile connection type uses a GSM/GPRS network. To study on the Ajax mobile performance over the 3G wireless network may gain much attention in the future. But, for now, our study is based on GSM/GPRS network, there are some reasons. First, the GSM/GPRS network coverage has more wide coverage than the 3G network. For New Zealand Vodafone, GSM network can reach 97% of New Zealander, 3G network covers 57% of New Zealander (Vodafone New Zealand, 2007). Second, 3G is not a wireless network technology name; it includes many different network technologies, e.g. WCDMA (Wideband Code Division Multiple Access) or refers to as 3GSM, EV-DO, UMTS, or HSDPA. There are too many 3G standard or technologies around the world and none of them are worldwide standard yet. Finally, 3G handset is more expensive than the GSM handset.

However, all the wireless networks are high latency network. GPRS link latency is very high, 600ms-3000ms for the downlink and 400ms-1300ms on the uplink. Round-trip

latencies are 1000ms or more (R. Chakravorty, Cartwright, & Pratt, 2002). In a lab test environment, CNN's (CNN.com) download time over GPRS using Mozilla (Mozilla, 2007) in a desktop computer is around 125 seconds and 170 seconds (Rajiv Chakravorty & Pratt, 2002).

But, the latency is not improving with increased bandwidth, and deploying a 3G mobile phone network is not very helpful (Hunaiti, Garaj, Balachandran, & Cecelja, 2005) e.g. TCP initial time on GPR is 5-7 seconds, but the TCP initial on 3GSM is up to 12-15 seconds (Nortel, 2007). The Ajax engine only fetches useful data when the application request, these request may be not continuously. Then, the mobile device may need to setup the TCP connection for each request; the long initial time may slow down the whole Ajax system.

So, we are using Vodafone New Zealand GPRS network as our test bed wireless network, because it is wildly used, more cheaper and has fast initial time.

2.6.3 Mobile Web browser

Nokia studies of S60 Smartphone users show browsing is generating over 60% of data traffic (Nokia, 2005). Today, hundreds of mobile phones are available on the market, and more and more mobile phones are offering mobile browsers. A mobile Web browser is a Web browser designed for mobile devices, such as mobile phones, PDAs, or PPC (Pocket PCs). These mobile Web browsers are device-oriented. They need to be optimized or modified for the devices they are going to run on, because different vendor's mobile phones have different kinds of hardware and embedded operating systems and applications.

There is no certainly standard for mobile development; each mobile phone or mobile device is unique. To illustrate, the size of the screen is different- N70 uses 176 x 208 pixels, 35 x 41 mm TFT, 256K colour screen, but Motorola RAZR2 V8 uses a 2.2" QVGA (320 x 240 pixels) internal display, 262K colour screen. Their display size and resolution are totally different. Not only is the hardware different for each mobile phone, but also the operating system is different. N70 uses Symbian OS 8.1a, Series 60 UI as its operating system; Motorola RAZR2 V8 uses Linux as it operating system. So, from hardware to software, there is no application or Web browser and both run on both mobile phones without any optimized or modified applications.

There are many mobile phones with portable Web browsers available on the market today, and some mobile Web browsers were developed to adapt to mobiles, such as Opera Mini, Opera mobile, NetFront, Nokia's S60 browser, Pocket Internet Explorer and so on.

Opera Mobile & Opera Mini

Opera Mobile (Figure 2.11) and Opera Mini are the Web browsers developed by Opera software. Both of them aim to allow mobile device user to surf on the Internet by using their mobile phone. Although, they all come from Opera Software, they have some differences.

Opera Mini is a thin-client-side application Web browser. All requests from Opera mini will go to an Opera remote proxy server, and pass through to the destination Web server. All the response from the destination Web server will reach the Opera remote proxy server first, and then the Opera proxy server will compress the Web content to reduce the size of data transfers, and enable a fast browsing experience at low costs even on simpler

phones. They call it Mini Technology which means Opera mini can only support whatever the proxy support. It can't support Ajax approach yet.



Figure 2.11: Opera mobile Web browser (multi-tabs)

Opera Mobile is a stand-alone Web browser that can run on your Windows mobile and S60 3rd edition mobile. It can access the Internet directly from your mobile phone and does not require any proxy sitting in the middle. Opera Mobile can fully support the latest and hottest technology on the desktop: CSS, JavaScript, Ajax, etc. and all of the features and the existing operator services based on the WAP 2.0 protocols.

Opera Mobile is the only true mobile browser to support both full Web browsing together with existing operator services based on the WAP 2.0 protocol.

Nokia's S60 mobile browser

Nokia's S60 browser is not the official name of this browser; we call it that because it is an embedded mobile Web browser for all the S60 3rd edition mobile phones. This Web browser aims to offer users a premium solution for mobile phone Web browsing. Its exclusive feature is the "minimap", which allows users to see the whole Web page as a small thumbnail (Figure 2.12).

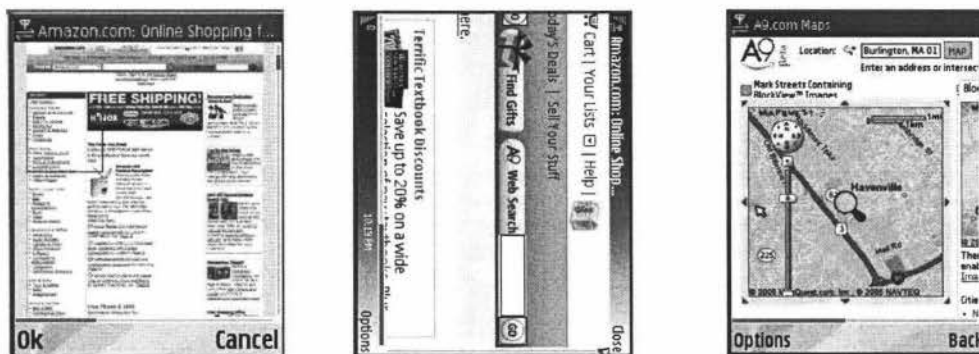


Figure 2.12: S60 Web mobile browser (minimap, full-screen, Ajax)

This S60 Web browser is based on the WebCore and JavaScriptCore components of Apple's Safari Web Kit. It can support HTML4.01; XHTML 1.0; CSS 1, 2, 3; DOM 1, 2; JavaScript 1.5 and the most important for our research – Ajax. Surprisingly, however the S60 Web browser did not work very well with our Ajax Webpage Website.

Microsoft Internet Explorer mobile

Microsoft Internet Explorer mobile is a Web browser that allows users to enjoy a rich Web experience as if they are using Internet Explorer (Figure 2.13) on their desktop. Internet Explorer Mobile supports most of the Web technology and it can connect to the Internet without any middleware or proxy to modify the Web content. Some base technologies support by IEM are SSL, HTML 4.0, XHTML, CSS, Cookies, and, of course, Ajax.



Figure 2.13: Microsoft Internet Explorer mobile browser

Other mobile Web browsers

There are some other mobile Web browsers that unfortunately do not support Ajax technology so far, such as, NetFront, Openwave, Teleca's Obigo, etc.

There are some considerations for choosing a mobile Web browser when implementing Ajax. The mobile Web browser must support JavaScript, especially XMLHttpRequest object. It also needs to support DOM, CSS and XML. On the other hand, the compatibility problem is very important in mobile devices. A test based on stand-alone mobile Web browsers is more useful than a special-made Web browser.

Opera Mobile, Nokia's S60 Browser, Microsoft Internet Explorer Mobile announced that they are supporting JavaScript and Ajax. More and more mobile browsers are announcing support for Ajax (Asynchronous JavaScript and XML); one of the major reason affecting the acceptability for Ajax at the beginning, the Cross-Web browser compatibilities problem, has been solved.

2.7 Alternative approaches to Ajax

As outlined before, the Ajax approach allows users to enjoy a rich Web experience on their Web browser; it also changes developers' habits, and a more user friendly interface will be applied to Website development. The advantages of Ajax look attractive for everyone; now the Ajax approach can be applied to more and more technologies, such as, Java, JavaScript, and ActionScript (Flash). These technologies can be used to create the Ajax engine, but still have some differences.

There is no thing or technology that is perfect for everything; every technology has its own strengths and weaknesses. Understanding the strengths and weaknesses for each Ajax technology can help us to decide the best approach for different Websites or Web applications.

2.7.1 Java/XML approach

Java Ajax means instead of JavaScript, using Java as the base language to create an Ajax engine to achieve the same features of Ajax, such as, asynchronous client-server communication and partial user interface update. There are a lot of resources for this Java Ajax approach (ajaxpatterns.org, 2007), AJAX JSP Tag Library(ajaxtags.sourceforge.net, 2007), JSON, Echo2, etc.

Strengths:

Java language is a binary execution, multi-threaded computing, and platform-independent object-oriented programming language. The natural programming language allows it to deliver good performance and complicate functionality applications, and these applications can be easy to maintain or update. The application can be adapted to other platforms easily. Team work is normal in object-oriented programming language development.

Weakness:

It requires JVM (Java virtual machine) for program running.

2.7.2 Macromedia's Flash and ActionScript approach

Before Ajax show up in 2005, Macromedia's Flash stood on the top of the user interface chain; it provided the best user Web experience by embedding SWF (SWF File Format Specification from Adobe documents the file format that plays back in the Adobe Flash Player.) in to the Web browser. It also provided similar features as Ajax, using Flash as a user interface and ActionScript as logic controller.

Strengths:

Very good and pretty graphics and animations can apply into the Website. Users can enjoy some wonderful user experiences including sound or video on Flash Websites.

Weaknesses:

ActionScript code is not a strict programming and design standard. It is hard to maintain is not designed for team development, and has performance and functionality limitations. For dynamic content generation, SWF is a pre-compiled format; there is no language that can dynamically change its content. Additionally, there is a lack of complete HTML, CSS support issues.

CHAPTER 3: LITERATURE REVIEW

Ancient saying: "The remembrance of the past is the teacher of the future".

Most of the mobile learning published papers are interesting in the basic concept of the mobile learning strategy; others are forces on the mobile learning model and framework, or how to create a good structure of mobile learning system. The technology-orient paper for mobile learning is not very popular in mobile learning domain.

Ajax is a new technology for Web developers, so published academic resources are limited. Mainly, these resources can be divided into two aspects, Ajax implementation and Ajax application performance measurement. Most of these articles focus on introducing Ajax, such as, what Ajax is, how Ajax works; or what Ajax can or cannot do. Some articles focus on how to implement Ajax technology into some particular software or technologies, for example, combining Ajax with encryption, or groupware, or GIS visualization Web services. Just a few articles are related to the Ajax performance, or an approach about how to compare the basic HTML and the new Ajax.

3.1 Mobile learning

The combination of wireless telecommunication and mobile computing is resulting in escalating transformation of educational world (Alexander, 2004). The growth and rapid evaluation of the wireless technology have created new opportunities anytime and anywhere learning paradigm (Seong, 2006).

The mobile learning concept is still developing, different defined announce by researchers. Pinkert, Hoppe, Milrad and Perez (2003) define it as an e-learning that uses

mobile devices and wireless transmission; Polsani (2003) define it as a form of education whose site of production, circulation, and consumption is the network; Traxler (2005), who defines m-learning as any educational provision where the sole or dominant technologies are handheld or palmtop devices; Sharples (2005) define it as a process of coming to know, by which learners in cooperation with their peers and teachers, construct transiently stable interpretations of their world.

Mobile learning is different from our traditional learning experience; it has its own problems and limitations. Mobile learner or Elearner may feel uncomfortable, because the learner cannot have face-to-face interaction with teachers or other students (Stodel, Thompson, & MacDonald, 2006). Mobile learning is mainly a time-constrained exercise with lightweight content-oriented instruction (Berri, Benlamri, & Atif, 2006).

And there are a lot of research in mobile learning domain (Attewell & Savill-Smith, 2003; Berri, Benlamri, & Atif, 2006; Caudill, 2007; JISC, 2005; Peters, 2007). All of these the mobile learning studies can be categorised to 6 aspects (John Traxler, 2007):

- Technology-driven
- Miniature but portable e-Learning
- Connected classroom learning
- Informal, personalised, situated mobile learning
- Mobile training /performance support
- Remote/ rural / development mobile learning

These 6 aspects and from the design process point of view (Berri, Benlamri, & Atif, 2006), there are three major categories: pedagogical learning models, adapt into mobile environments and the technically problem. Current, most of the mobile learning studies are creating a framework, pedagogical learning models, or learning portals.

These are some useful mobile frameworks and model designs (Brown, Ryu, & Parsons, 2006; Nakahara, Hisamatsu, Yaegashi, & Yamauchi, 2005; Mike Sharples, Corlett, & Westmancott, 2002).

The technical support for the Web-based mobile learning has been sparsely explored. A mobile learning system using “Instant Messaging” and wireless network (GSM/GPRS and 802.11 Wi-Fi) is developed by Kadirie (2007). This system is Web-based system the kernel of the system Instant Messaging server is created by Java. Learner uses mobile Web browser to login to this learning system to chat or view other learners’ messages.

And other mobile learning technically paper is from Seong (2006), He developed a mobile learning course manager portal to demonstrate and exemplify the usability guidelines proposed. Another Java + XML mobile learning example is from Kukka and Ojala (2006), They are using a real database for Digital Oulu Cultural developed a Web-based application that allow user use the database on both desktop and mobile devices. And in the end they conducted a usability test for the mobile system and the users have found their system is useful, usable and suitable for mobile devices based on the gathered data.

A Web-based collaborative learning sites has the bulletin board system (BBS) and allow learners to interact, exchange information, engage in discussion, and collaborate on projects (Nakahara, Hisamatsu, Yaegashi, & Yamauchi, 2005). Other good technical examples are (Cao, Tin, McGreal, Ally, & Coffey, 2006; W.-P. Lee & Lu, 2003; Theng et al., 2007). They are all using the Web-based technologies plus the mobile Web browser or mobile application to develop their mobile learning system.

These Web-based mobile learning systems used traditional synchronous client-server application architecture to develop the mobile learning system. These mobile learning systems usually using the GSM/GPRS network for long distance mobile learning, their system performances are based on the GPRS network status. But in most case, the Web application performance over GPRS network is not good (R. Chakravorty, Cartwright, & Pratt, 2002; Rajiv Chakravorty & Pratt, 2002; Stuckmann, Ehlers, & Wouters, 2002), as long as the GPRS network is a high latency wireless network and the Web application is using the “synchronous request/response model”. To improve the mobile learning system performance needs to update the entire GPRS wireless network or to apply some new approach into the Web-base mobile learning system. Ajax approach may be a good idea.

3.2 Ajax Implementation

Just after Garrett’s essay, the Ajax concept was widely accepted by Web application developers and Web designers. Today, researchers are interested in how to implement Ajax in to the real Web application realm. Are there any quantifiable benefits for end users and business, which include improved usability and faster applications?

Lerner published a series of articles called “At the Forge”(Lerner, 2006a, , 2006b, , 2006c, , 2006d, , 2007a, , 2007b); this series begins with an article, “Beginning Ajax”, that teaches you how to implement the Ajax technology into a Website. It is good for the beginner and outlines the important things for Ajax development and what kind of problems you may face during the Ajax Website development.

In real Ajax Web application development, Ajax developers are facing challenges. Rosenberg (2007), lead interaction designer for the new Yahoo mail system, shows us some drawbacks, when they were developing this new mail system. The first drawback is

the progress and activity indication. For the old style Webpage, the Webpage is either completed or the page is loading. But in the Ajax environment, after the whole page download is completed, it still can be updated part by part. In this scenario, telling users what the Webpage or JavaScript is doing is important, especially if the update will take an unpredictably long time (Rosenberg, 2007). Rosenberg and his team developed an asynchronous progress spinner to solve this indication problem. The spinner will note to the user what the JavaScript is doing in the background.

The second drawback is related to the “back” button (Rosenberg, 2007). For some record, about 40% of clicks on the traditional Webpage clicks on the “back” button, when they want to go back to the previous page. However the “back” button does not work as it used to in the Ajax platform. They are still working on the “back” button solution. On the other hand, a bible of Ajax “Ajax in Action” suggests a solution for the “back” button.

Other articles focus on how to implement Ajax in some particular applications, not only the Web application, but also the desktop application. Some researchers from Indiana University (Sayar, Pierce, & Fox, 2006), improved normal GIS (Geographic information systems) by using the Ajax model, compared with the old GIS, the new Web-based GIS provides high usability and performance increases.

Network security is becoming more and more important these days; most of the researchers are looking at the threat from viruses, spyware and online fraud. Ryan and Reid (2006) developed encryption email agents offering users desktop-like interaction experienced by the Ajax model (Ryan & Reid, 2006). This system overcomes the usability problems for normal email systems. “Ajax Security in Groupware” is a paper evaluating the Ajax platform, that came up with some solutions for a special focus on the context of groupware (Sonntag, 2006).

Before the Web or desktop development, a good architectural style is very important, especially, migrating a multi-page style Web application to one-page multi-update Ajax Web application (Mesbah & Van Deursen, 2007a). Framework can help developers to avoid the complexity of Ajax development and provide a platform for developers, such as, Web browser compatibility, process/thread racing, user indication problems and so on. There are some frameworks for this: Echo2, GWT (Google Web Toolkit) (Goth & Costlow, 2007), BackBase, and SPIAR (Mesbah & Van Deursen, 2007b).

An instead of implementing the Ajax technology is implementing an Ajax function into the Web browser would be an interesting idea. Researchers from Japan propose a new Web browser can partially update a Web page, also supporting asynchronous communication with Web server computers like the Ajax approach (Hanakawa & Ikemiya, 2006). They apply the Ajax approach with asynchronous communication model to a Web browser instead of the Web application itself. Their Web browser can improve the operation that is equivalent to the operation of desktop applications without revising program codes of the Web application. It also can partially update the Web page even though the Web application does not adopt the Ajax approach.

3.3 Ajax performance on the desktop

All Ajax implementation articles are telling us that Ajax can significantly improve the Web application performance. But none of these articles are experimenting or measuring the comparative performance between the traditional HTML approach and Ajax approach. Then there is a question mark, when we are developing Ajax base application. Ajax, is it that good?

Previous research suggests that Ajax can significantly improve web application performance on the desktop. One commonly cited Ajax performance evaluation is the

one reported by White (White, 2005). In his study, the Ajax application transferred on average just 27% of the bytes that were transferred by a traditional HTML application. Not only was there an improvement in the transferred byte volume, but there was also an improvement in performance, a 68% overall improvement in data transfer time. But these two applications used in this study, however, did not have the same user interface. The users' skill levels and training were also not assessed in the report, and these factors may have affected the experimental outcomes.

A more controlled Ajax performance measurement comes from Smullen and Smullen (2006). They compared the client-side performance of a real-life HTML application and an Ajax application that implemented the same user interface. In this case, the variable of training and users' skill level can be ignored. Experimental data was collected on the performance of each when presented with the same set of 110 tasks. Performance measures were computed for the HTML application and for the Ajax application.

Later they extended their study by collecting data on a statistically significant sample size and included server performance results. Response size and service time performance measures computed for the applications provided significant performance improvements in response size for the Ajax application (56%), thereby reducing bandwidth requirements. Ajax provided a mean service time improvement of approximately 16% (Smullen & Smullen, 2007).

Another paper about the Ajax performance on desktop environment is from University of Rome Tor Vergata, using Ajax approach to developed a Web-chat application, and compared the Qos between a traditional client-server web chat model and the asynchronous one implemented using the Ajax model (Angelaccio & Buttarazzi, 2006). In particular, they have forced values greater than 10 chat-messages per second in the AJAX client by obtaining as saturation level in correspondence of values 18-20

chat-message/sec. Ajax application faster than traditional application by reducing the unnecessary network traffic.

Although Ajax performance has been measured on the desktop, there is no currently available performance measurement research about Ajax on mobile devices. Therefore we do not know if it can improve mobile Web-based application performance, or indeed if Ajax is effective in the mobile environment. Therefore this study focuses on measuring Ajax performance in the context of the mobile Internet.

In summary, all of these Ajax technologies or concepts are measured based on desktop computer environment. Ajax is a client-side technology which means when the platform moves from the desktop to your mobile devices, as long as the client platform is changed, the results may be different.

3.4 WWW performance over GPRS network

Huge amounts of resources relating to mobile technology and mobile application development has been done which is going to make our mobile devices more powerful to do more work for us, and to make our lives easier.

Most of the research is related to the application on the mobile devices, such as enhancing an E-commerce course by using WAP programming (Rahul, 2001), an online instructional testing in a mobile environment (Cerise, Gail, Terry, & Nelson, 2003), and even and gaming system for 3G-Networks (Akkawi, Schaller, Wellnitz, & Wolf, 2005). But all these developments are based on simulated mobile network environments. In fact, the GPRS still need to do some modifications to give mobile Web applications better support.

A lot of work has been done for the mobile phone network to make it more efficient. Some papers analyzed the performance of GSM/GPRS mobile phone network performance, and came up with solutions to increase GPRS network performance by using the Multi-slot (Zhang & Soong, 2004). Another way to improve the GPRS is by using the dynamic simulator models. In their study, a set of optimized and tuned parameters are suggested, which could help operators to go towards more aggressive network deployments with more advanced RRM schemes and at the same time provide higher quality of service (QoS) to the end users with maximized capacity (Che, Hamalainen, Ryyanen, & Moisio, 2003). And to estimate the GPRS performance, one of the approximation methods is introduced by Shaoji Ni and Sven-Gustav Haggman; this method is used for evaluating the GPRS performance of single-slot service in the variable radio resource (Ni & Haggman, 1999).

But the WWW performance over real-life GPRS network is very disappointing for most of the users and researchers, especially, TCP/IP Internet protocol performance over GPRS network. TCP protocol needs to initialize before it starts to transfer actual data; the process takes over 7 seconds to expand the congestion window sufficiently to enable the connection when the initial connection request was made (Rajiv Chakravorty, Clark, & Pratt, 2003).

In short, the current mobile network environment is not very good or acceptable for the mobile Web application users, some works still needs to be done to the mobile Web application for users to really accept the mobile Web application.

CHAPTER 4: METHODOLOGY

Ancient saying: "Deliberate in counsel, prompt in action."

This chapter outlines the design research method which is used in the research, describes what it is and why we use the method, and the process of using the method in this research.

4.1 What is the design research method?

Design — to invent and bring into being; it means creating something new that does not exist in nature (Vaishnavi & Kuechler, 2007). The general design research process is also the process generating and accumulating knowledge process. As Owen (1997) said:

Knowledge is generated and accumulated through action. Doing something and judging the results is the general model . . . the process is shown as a cycle in which knowledge is used to create works, and works are evaluated to build knowledge.

The design research method process has 5 basic steps: awareness of problem, suggestion, development, evaluation, and conclusion (Figure 4.1)

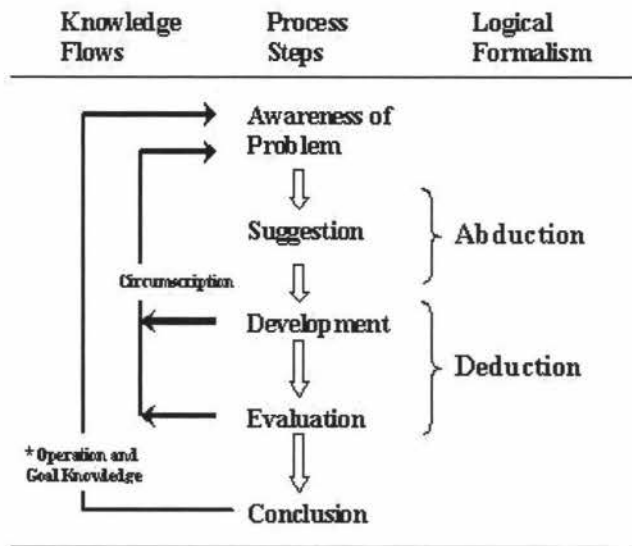


Figure 4.1: Reasoning in the Design Cycle

In this model all designs begin with awareness of a problem. Design research is sometimes called “Improvement Research” and this designation emphasizes the problem-solving/performance-improving nature of the activity. Suggestions for a problem solution are abductively drawn from the existing knowledge/theory base for the problem area (Pierce, 1931). An attempt at implementing an artefact according to the suggested solution is performed next. This stage is shown as “development” in the diagram. Partially or fully successful implementations are then “evaluated” (according to the functional specification implicit or explicit in the suggestion). “Development, evaluation and further suggestions” are frequently iteratively performed in the course of the research (design) effort. The basis of the iteration, the flow from partial completion of the cycle back to awareness of the problem, is indicated by the circumscription arrow. Conclusion indicates termination of a specific design project (Vaishnavi & Kuechler, 2007).

4.2 Why use the design research method?

According to the March and Smith (1995), there are four outputs for the design research method: constructs, models, methods, and instantiations. Later, Rossi and Sein (2003)

and Puroo (2002) have set forth their own list of design research outputs. However, they can be mapped directly to March and Smith's list as fifth output, better theories (Table 4.1).

Table 4.1: The outputs of design research

	Output	Description
1	Constructs	The conceptual vocabulary of a domain
2	Models	A set of propositions or statements expressing relationships between constructs
3	Methods	A set of steps used to perform a task – how-to knowledge
4	Instantiations	The operationalization of constructs, models and methods.
5	Better theories	Artefact construction as analogous to experimental natural science

In this research, the main target is to measure the performance of the Ajax enabled Website on mobile devices over the TCP/IP GPRS network. Ajax is the latest approach that can significantly improve the Website user interaction experience and reduce the transmission usage (Smullen & Smullen, 2007). But all the performance measurements are based on desktop computers or PCs; they did not measure the performance on mobile devices, especially on mobile devices over the GPRS mobile phone network.

In order to measure the Ajax performance, we need to develop an Ajax enabled Website and a non-Ajax Website first, then, measure their performance over a GPRS network on mobile devices. It means a better theory will be created and measurement on this output fit the output of design research. A conclusion will address in the end whether mobile Ajax provides measurable advantages over non-Ajax mobile applications.

4.3 Apply design research method process in this research

For this research, we are not only applying Ajax approach to a mobile learning system, we are also interesting in the performance difference between the Web-based Ajax mobile learning system and the traditional Web-based mobile learning system. We are doing iteration once with two evaluations (technical performance measures and usability test). These mobile learning system performance measures evaluation results can be used in the further research.

According to the design research process, there are 5 steps of this Ajax performance design research, and each step has its own output (Figure 4.2). They are awareness of problem, suggestion, development, evaluation (quantitative and qualitative), and conclusion.

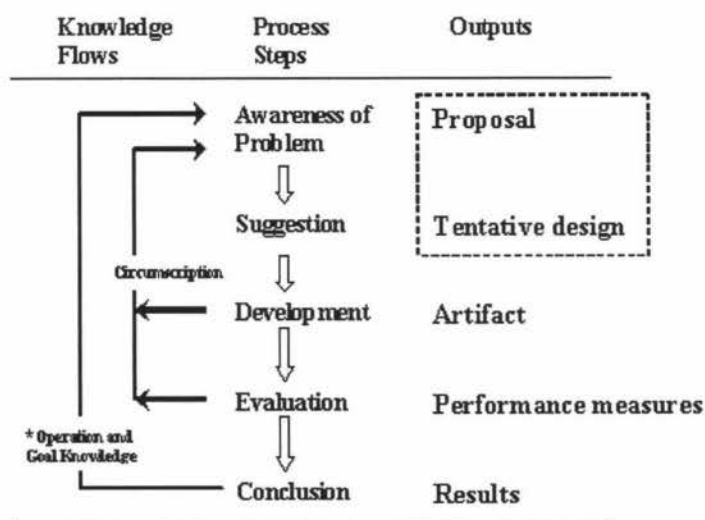


Figure 4.2: The General Methodology of Design Research

Awareness of problem

In the past few years, more than one billion people used the Internet, and that is about two times the amount of usage in 2000. Africa and Middle East recorded exceeding 400% increases (Internet Usage Statistics, 2007). Meanwhile, worldwide mobile phone

sales totalled 816.6 million units in 2005, a 21 percent increase from 2004 (Gartner, 2006). In 2006, this number surpassed 990.8 million (Gartner, 2007). As the amount of Internet and mobile users increase, more and more mobile phone vendors are integrating Internet access functions into their mobile units. Mobile Internet access accounts today for only a small proportion of Internet users. However, wireless Internet access devices are becoming increasingly popular. In the business realm, Internet and mobile technology can also help companies keep in touch with travelling staff by Internet and email

On the other hand, worldwide mobile phone sales totalled 816.6 million units in 2005, a 21 percent increase from 2004 (Gartner, 2006). In 2006, this number surpassed 990.8 million (Gartner, 2007). Using GPRS network to access Internet on their mobile device become more and more popular.

But, using mobile phones to surf the Internet is different from using desktop or laptop computers. Due to the limitations of mobile phone Web applications and hardware weaknesses, people cannot enjoy Web surfing in wireless mobile phones due to small screens, high network latency, small bandwidth, interface complexity and so on. What can we do to improve the mobile Web user's experience?

Suggestions

In 2005, Garrett introduced us to a new Web approach, Ajax. After that a lot of researchers and developers began using the Ajax approach to develop their new generation Website or Web applications on desktop. As outlined in the literature review, most of the articles and papers only focus on how to implement the Ajax approach in their research, and stated that Ajax can improve the Website or Web application performance, but did not provide any evidence or measurement process to support their thesis. Only a few papers from Smullen and Smullen (2006; , 2007) provided the measurement process for Ajax to support that Ajax application performance is better than

traditional application on the desktop. All of these researches are based on the desktop environment and using wire-cable as the connection medium, the Ajax Web-based application performance over the wireless network still uncover.

Ajax can significantly reduce the transmission volumes and improve the user's experience on the desktop environment. Meanwhile, Web-based mobile application needs some solutions to reduce the transmit volumes and provide better user experience. Can we deploy the Ajax approach to the mobile environment? If we do, what is the result? We expect that an Ajax Web-based application provides acceptable user experience for mobile Web-based application users, e.g. Web-based mobile learning system user, and reduce the wireless network transmission volumes and frequency.

Development

Two mobile quiz learning Web applications will be developed by Ajax approach and non-Ajax approach. The first Web-based mobile quiz learning system is Ajax enabled Web application; the second Web-based mobile quiz learning system is using the traditional ASP (Active Server Pages) Web technology to develop.

Although these two Web applications use different technologies, they have the same user interface. We are interesting in the performance of these two Web applications, to avoid other ingredients to affect the system performance, such as user interface. Using the same user interface, to avoid the risk of training for different user interfaces or confusing the users is necessary. And they can be accessed from the same device, from desktop computer or mobile devices (e.g. mobile phone, PDA, Pocket PCs).

So, there are some requires for these Web-based mobile learning system. First, they must have the same user interface. Second, the learning process must the same including the

same questions and answers. Third, they use different technologies. Finally, they can be access from the mobile devices over the GPRS network.

Evaluation

After the development, we will evaluate these two Web-based mobile learning application performances from three aspects: theoretical evaluation, real-life performance evaluation, and usability testing.

Theoretical evaluation (quantitative study)

This means emulate the whole user's visit in the Website process and calculate all the data on paper to find out whether these systems are fitted into our design (the same interface, the technology is different, etc).

First, we assume there is a user visiting the Website or Web application by Web browser, according to the development information, and predict the data transfer usage between Web server and mobile Web browser. Then, compare the ASP Web application's data with Ajax enabled Web application's data. According to these data, modify or improve the Website; this can be considered a tentative evaluation.

Real-life computer logging evaluation (quantitative study)

Set up a test bed (mobile device – GPRS network – Web server) for real user visit action, and upload the Web application into Internet Web server. Test these Web applications over GPRS network on mobile device and collect all the data transfer information data between Web server and client from Web server log. Finally, we analyse data from the log to evaluate the Ajax and non-Ajax Web application performance over the GPRS network.

User evaluation - observational usability test (qualitative study)

A small group of users will be invited for an observational usability test. “Cooperative evaluation”, a variation of “think aloud” observational techniques will be implemented in this small group of usability tests.

They will be asked to visit both Ajax and non-Ajax Websites on different performances (I-mate SP5 mobile smart phone), and they can say anything related to Websites when they are visiting. All comments will be collected and analysed, especially the comments for the Website performance.

This user evaluation attempts to answer this question: Does the user think Ajax is better than non-Ajax M-Learning Website? Which one do they prefer? If Ajax is better than non-Ajax, is it significantly improving the Website performance the same as the real-life computer logging evaluation, or just slightly improved the performance, too small to be noticed by the mobile Web Internet users.

Conclusion

After the evaluation, the performance of the Ajax Web-based mobile learning system is very clear now. Combining the three aspects (theoretical, real-life, usability) of the evaluation provides a conclusion for this research and outlines the limitation of this study.

4.4 User evaluation design

To evaluate the Ajax Website and non-Ajax Website performance over the mobile devices, we implemented within-subject design as our experiment design methodology, using the “cooperative evaluation” and the “post-task walkthrough” as our observational techniques.

4.4.1 Within-subject design

Because the goal of this user usability test is to find out “Can Ajax Web application provide better Website performance that good enough to notice by end-client users?”, then, first we need to measure Website performance, Is Ajax web application better than the ASP web application or equal the same. Second, we also need to measure the preference data to avoid the new technology have to use by users. That is about which system user prefer, the Ajax or the ASP.

For a small group usability test and the requests of your evaluation, the with-in subject design is suitable for our usability test. In a within-subjects design, the values of the dependent variable for a task or a set of tasks are compared with the values for another task or another set of tasks within one participant's data. In other words, within-subjects designs are designs in which the same variable is measured repeatedly on the same participant under different task conditions (Ergosoft Laboratories, 2003).

The advantage of within-subject design is, first, need fewer numbers of participants; second, and can get the preference data and the get comparison data. Applying within subject design in this usability test, participants need to complete the same task on both Websites. User’s task is finish all 5 multi-choice questions on the Website, the questions are equated in difficulty (Appendix 7).

Since, users have experience on both Websites; they can compare these two Websites and make some statement. Meanwhile, we can collect the performance data (Task completion time). To avoid user transfer learning from the first Website, we provide user a Demo Website with the same interface but different questions that help user to get familiar with the test device and the task process.

4.4.2 Observational Techniques

A popular way to gather information about actual use of a system is to observe users interaction with it. Think aloud protocol is a method used to gather data in usability testing in product design and development, in psychology and a range of social sciences. The think-aloud method was developed by Clayton Lewis(Lewis & Rieman, 1993).

Think aloud (also spelled think-aloud) protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling, as they go about their task. This enables observers to see first-hand the process of task completion (rather than only its final product). Observers at such a test are asked to objectively take notes of everything that user's say, without attempting to interpret their actions and words. Test sessions are often recorded and videotaped so that developers can go back and refer to what participants did, and how they reacted. The purpose of this method is to make explicit what is implicitly present in subjects who are able to perform a specific task.

The 'Cooperative Evaluation' is a variation of 'Think aloud' observational techniques. The advantage of cooperative evaluation is simply, and encouraged the user to be a collaborator in the evaluation. It allows user to ask the evaluator any question if a problem arise. The relaxation environment have some advantages, first, it can reduce the user's finish time, easier for user to learn the test process from the conversation. Second, ask any question any time the evaluator want, it can encourage participants to express themselves, when they are doing the task. Third, user can identify the problem area very quickly, maximize the effectiveness.

The type of record for evaluation in 'Think aloud' and 'observational evaluation' is known as a 'protocol'. These 'protocol' include paper and pencil, Audio recording, video recording, computer logging, and user notebooks (Dix, Finlay, Abowd, & Beale, 1998). Paper and pencil plus video recording are applied in this usability test. Video recording

can record all user actions, when they are visiting the Website, these data can be study later.

But only paper record may not good enough to provide information for later study. We can observe the user's action and see the action user is doing, but most of time we don't know why they are doing that, these observation data lack of interpretation. In that case, the post-task walkthrough need to be implemented, it reflecting back to the subjects their actions after the event. The advantage of a delayed walkthrough is the analyst has had time to frame suitable questions and focus on specific incidents (Dix, Finlay, Abowd, & Beale, 1998). In our case, post-task walkthrough can help user notice some performance problems.

CHAPTER 5: IMPLEMENTATION

This study is based on a controlled measurement of data transfer volumes and times, comparing an Ajax Web application with a similar non-Ajax implementation. We developed two IQ tests (multi-choice) about Web applications with the same interface and functionality (Figure 5.1), to compare the performance of an ASP based Website and an AJAX based Website in a GPRS mobile network environment. We collected the experimental data from the server and mobile phones, including bandwidth usage, response times, and mobile unit storage requirements.

The experiment aimed to measure the performance of Ajax Webpage Websites on mobile phones. First, we predicted the Website performance based on paper calculations, theoretically analyzing the data volumes and response times over the WAP 2.0 protocol. Second, we analyzed real-life data collected from the system log of the Web server. Finally, small group of usability tests were conducted to find out if the users noticed the difference between the Ajax and non-Ajax Web applications.

There are many Ajax frameworks available now, but most of them are designed for conventional desktop/laptop environment, not suitable for mobile environment. In the tentative design, I tried to use Microsoft Ajax 1.0 framework as the base of the whole application. But the whole Web is too big to use in mobile devices. For the same mobile phone learning Website, our demonstration study Website is just about 10Kb (except Database), but if the system want to achieve the same functions by the framework, it will close to 100Kb. It is very inefficient. So, this Ajax mobile quiz learning system is not using any framework.

5.1 The Mobile learning application

In order to compare the performance of a real-life HTML application and an Ajax application two mobile learning Web applications were created: an Ajax Web application and a similar non-Ajax implementation (ASP). They have the same user interface and functionality (Figure 5.1).

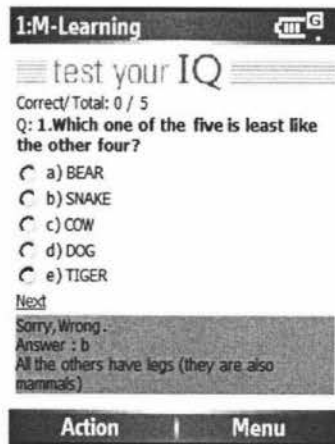


Figure 5.1: Mobile Learning interface (Opera Mobile)

The mobile phone Web application was based on a three layer architecture: the mobile phone user interface (Web browser), server pages and a database server. The Web server was Microsoft Internet Information Services (IIS), the server pages were written using ASP 2.0 and the test database was stored in Microsoft Access. All dynamically generated Web pages were valid XHTML 1.1, compatible with the mobile browser (Opera Mobile) used in the test. All data transfer was through the Vodafone New Zealand GPRS (GPRS/GSM 900/1800/1900MHz) network by Class 2 “2+1” multi-slots (2 downlink, 1 uplink channels).

The system logs were activated in the IIS server, so all traffic going to and from the Web server was recorded. This Webpage was used to isolate and monitor all the transfer

(inbound and outbound) between the Web server and mobile terminal. The collected data was used to analyze and measure the Websites' performance over the GPRS network.

In this study, the Ajax Web application still is an asynchronous communication model; it means the processing does not wait until the response returns from the server.

5.1.1 ASP mobile learning system

The ASP Web application only contains three ASP pages: a start Webpage, a question display Webpage, and a finish Webpage. These pages interact with the database to provide page content. Other supporting files are an image file (in GIF format) and a JavaScript file to manage the question fetching process. When the user clicks on the submit button, it activates a JavaScript function that sends a request to the server to fetch a new question or answer. All the files storage sizes on the Web server are shown as following table (Table: 5.1).

Table 5.1: ASP Web application files size

File Name	Size(bytes)	Detail
index.asp	818	Start Webpage
mlearning.asp	3010	Display questions
iq_header.gif	1470	Header image file
asp_m.js	1530	JavaScript file
hello.css	45	CSS file
learning.mdb	496000	Questions database
end.asp	744	Finish Webpage
Total size	504309 bytes	

The ASP Web application is developed in the traditional way. An ASP file includes CSS and JavaScript file path in its header. It also contains the Webpage layout information, Web content, and ASP code to dynamically generate the HTML Webpage. The ASP code also provides the verification function to verify the user's answer, once the user submits the answer.

The JavaScript (Asp_m.js) file only has two main functions, first is to fetch the next question for the user; the question number comes from the URL. The second main function is to validate the user's submission to avoid empty or error submissions.

The CSS is used to manage the whole Webpage's front size, background colour, answer layout and its layout.

Of course, the database file is used to save all the mobile learning multi-choice questions. In the question database, there is only one question and answer table, all questions, answers and explanations are included in this table.

The ASP Web application has the same user interface as the Ajax Web application and the same image file (iq_header.gif), same CSS file (hello.css) and shares the same database.

5.1.2 The Ajax mobile learning system

The Ajax application version is similar to the ASP Web application. On the client side, it includes three HTML pages: a JavaScript file, a GIF image file, and a CSS file. There is also an ASP file and an access database running on the Web server. The ASP file used in the Ajax application is different from those in the ASP application. It is used as both a

database connector and XML file generator, because the communication between client and server is via XML (Table 5.2).

Table 5.2: Ajax Web application file sizes

File Name	File Size(bytes)	Detail
majax.js	4750	JavaScript file(Ajax Engine)
Hello.css	45	CSS file
2.asp	1040	Database connection & XML generator
end.htm	1530	Finish Webpage
index.htm	526	Start Webpage
iq_header.gif	1470	Header image file
learning.mdb	496000	Questions database(same as asp version)
mlearning.htm	1520	Display questions
Total size	506836 bytes	

The total size of ASP Web application is about 2K bytes smaller than the Ajax version. The major difference between them is the JavaScript file size. The ASP Web application JavaScript file is only 1.5Kb; the Ajax Web application JavaScript file is almost 5Kb. That is because the Ajax Web application JavaScript file not only contains the same functions as the ASP version, but also has some logical functions, such as verify user's answer.

Ajax as a new Web development approach, the purpose of its files is different from the ASP Web application file. Explaining the difference between the Ajax and ASP Web application file can help us understand the Ajax approach better.

In the Ajax Web application, the HTML file size is very small because it is only used to lay out the main structure for the Website, such as, where to display the questions or where to display the answers and explanation. The most important part of the HTML file

is that it provides an entry page for the M-learning Website and the JavaScript file path (Figure 5.2). The HTML can be considered as a content template.

The JavaScript file, which is acting as an Ajax engine is very important for the Ajax Web application and provides most of the functions for the Website. There are two main aspects and five basic functions provided by this JavaScript file (Ajax engine). The first part of the Ajax engine is used to asynchronously transfer the data between the client Web browser and server. The second part is used to manage the client side user interface, and a logical function used to verify the user's answer.

```

<link rel="stylesheet" type="text/css" href='hello.css' />
<title>M-Learning</title><head>
<script src="majax.js"></script>
</head>
<body onLoad="start()">
<table width="75" border="0">
.....
    <td><form name="form2" id="form2" action="#" onSubmit="return false;">
        <input type="radio" name="answer_radio" id="answer_radio" value="a" >
        a) <span id="a"></span> <br>
        <input type="radio" name="answer_radio" id="answer_radio" value="b" >
        b) <span id="b"></span> <br>
.....
        <input type="radio" name="answer_radio" id="answer_radio" value="e" >
        e) <span id="e"></span>
    </form><span id="sbutton"></span></td>
</tr>
<tr>
    <td>
    <div id="answerfield"><span id="answer"></span><br>
        <span id="explanation"></span>
    </div></td>
</tr>
</table>

```

Figure 5.2: Ajax Web application HTML file

Function “GetXmlHttpRequestObject()” loads first to make sure the client Web browser can support the XMLHttpRequest object or not (Figure 5.3). Function “showQuestion” will be triggered when the user starts to submit a request; it is used to submit user requests. When a response returns from the server, “stateChanged()” can help create a new Webpage for the user interface.

```
function GetXmlHttpRequestObject()
{
var xmlhttp=null;
try {
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e) {
// Internet Explorer
try {
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e) {
try {
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
Catch (e) {
xmlhttp=null;
}
}
}
return xmlhttp;
}

function showQuestion(){
.....
if (xmlhttp==null){
alert ("Your browser does not support AJAX!");
return; }
.....
}
```

Figure 5.3: Function getXmlHttpRequestObject (GetXMLHttpRequestObject())

When the Web browser supports an XMLHttpRequest object, then the key Ajax function starts. “ShowQuestion()” function will be created as an XMLHttpRequest object instance. A request is made to the server by open() and send() method (Figure 5.4). The open() method is a one of XML Http Request operations. It is used to set up or initialize a

request to the server. The `send()` method actually sends the request to the server. The “`onreadystatechange`” properties will be triggered at every stated change. It means the `stateChange()` function will run in every stated change.

```
xmlHttp=GetXmlHttpRequest()
...
xmlHttp.open("GET",url,true);
xmlHttp.onreadystatechange=stateChanged; //stateChanged() run in every state change.
xmlHttp.send(null);
```

Figure 5.4: XMLHttpRequest object instance (ShowQuestion())

When the request is sent to the server, in fact there is only one variable that sent to the server from the client, a question number. The question number variable is submitted in “`get`” method by JavaScript. The question number is used as a parameter to search in the database (Figure 5.5).

```
dim var_id
var_id=request.querystring("q")
sql="SELECT * FROM QAE WHERE ID=" & Cint(var_id)
```

Figure 5.5: Query Database (2.asp file)

ASP code only can run on the server side. When the query result is obtained from the database, the ASP file will dynamically generate a text based XML file, according to the search results from the database (Figure 5.6). In this case, just a few characters need to be returned to the client Web browser. The data stream only sends out data; it does not include any page layout information.

```
response.write("<?xml version='1.0' encoding='ISO-8859-1'?>")
response.write("<question>")
response.write("<qu>" &rs.fields("question")& "</qu>")
response.write("<a>" &rs.fields("a")& "</a>")
response.write("<b>" &rs.fields("b")& "</b>")
...
response.write("<e>" &rs.fields("e")& "</e>")
response.write("<an>" &rs.fields("answer")& "</an>")
response.write("<ex>" &rs.fields("explanation")& "</ex>")
response.write("</question>")
```

Figure 5.6: XML file generator (2.asp)

Back to the client side, when the Ajax engine receives the XML file response from the server, the “XMLHttpRequest.readyState” value equals “4”, which means the request is complete. Once all the data is ready for the Web browser, the codes will update the user interface based on the latest information.

A variable xmlDoc is created and used to handle the XML file string. In the HTML file, the tag and unique tag ID show the Ajax engine and the content update location. The innerHTML statement can partially update the HTML page by modifying the content of the tag. As the code in Figure 5.7, just the content of the “question” span tag will be updated by the content of the “qu” tag from the XML string, which is the multi-choice question.

```
function stateChanged()
{
    if (xmlHttp.readyState==4)
    {
        var xmlDoc=xmlHttp.responseXML.documentElement;
        document.getElementById("question").innerHTML=xmlDoc.getElementsByTagName("
qu")[0].childNodes[0].nodeValue;
        .....
        answer=xmlDoc.getElementsByTagName("an")[0].childNodes[0].nodeValue;
        explain=xmlDoc.getElementsByTagName("ex")[0].childNodes[0].nodeValue;
    }else{
        document.getElementById("question").innerHTML="Loading...";
    }
}
```

Figure 5.7: Partial update interface

As shown in Figure 5.7, the Ajax Web application downloads the questions, multi-choice options, answers and explanations at one go. As mentioned before, the Ajax engine can contain some logic functions; here the Ajax will verify the user's answer on the client-side, instead of sending the answer back and waiting for another time (Figure 5.8).

```
function verify(){
    var tempa=getCheckedValue(document.forms['form2'].elements['answer_radio']);
    if (tempa!=""){
        if (tempa==answer){
            ...
            document.getElementById("correct").innerHTML=correct;
            document.getElementById("answer").innerHTML="Right!";
        }
        else{
            document.getElementById("answer").innerHTML="Sorry,Wrong. </br> Answer : " +
            answer;
        }
        document.getElementById("explanation").innerHTML=explain;
        //apply "explain" style to "answerfield", "explain" style defines in CSS file.
        document.getElementById("answerfield").className='explain';
        document.getElementById("sbutton").innerHTML="<a href=\\"javascript:showQuestion()\">Next...</a>";
    }
}
```

Figure 5.8: Verify() function

A brief description for the ASP mobile learning Web application and Ajax mobile learning Web application is done. Because these Web-based mobile learning applications are using the traditional Website quiz user interface, it is easy to use them.

5.1.3 Using the mobile learning system

The biggest advantage of these quiz mobile learning Web application is they are Web-based, user don't need to install or setup them up before using them. As long as the mobile device embedded Ajax mobile Web browser, user can use this quiz mobile learning system. Besides, these Web-based learning applications can be also access from desktop computer.

Because the Ajax and ASP mobile learning system are the same for everything except the technology, these systems are using in the same way. Figure 5.9 shows some screenshots for the quiz mobile learning systems. It begins with a welcome page (1). A hyperlink allows you to start their learning. There are total 10 questions need to answer for each system, you can choose your prefer answer by rolling the navigation button on the mobile phone or just select it on the PDAs touch screen (2). Once you confirm your answer, you can select the “Answer” to find out the correct answer. The correct answer will display under the questions with the explanation for the answer in light blue frame. Meanwhile, the “Answer” link will change to a “Next” link for next question (3). When you finished all questions, a finish page will display and allow back to the start page (4).

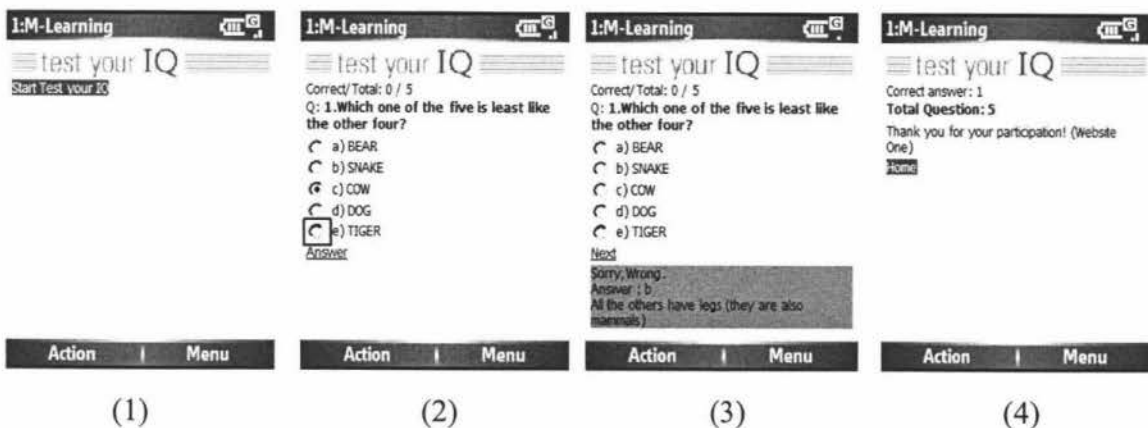


Figure 5.9: Using the quiz mobile learning Web application (Opera Mobile)

The mobile devices usually have small screen, display more questions in one screen may increase the loading speed, but it also may cause user trouble to keep roll up and down the navigation button to read the questions. On the other hand, simple user interface design can help us really focus on the technology runs behind the screen, and easy for the user to feel the difference. So, the quiz mobile learning system only displays one question at a time.

Next section will start the evaluation for the whole mobile learning system. Before the real-life performance measurement, some paper predictions will apply in the next section.

5.2 Theoretical performance

In this section, we predict the Website's performance based on application analysis and paper calculations, theoretically analyzing the data volumes and response times over the WAP 2.0 protocol.

5.2.1 Theoretical ASP system performance

For the ASP mobile learning Web application, when the Web browser requests an ASP file, the Web server passes the request to the ASP engine. The ASP engine reads the ASP file, line by line, and executes the scripts in the file. Finally, the ASP file is returned to the browser as plain HTML. These plain HTML files also contain all elements including header, content and layout. So, we can predict the file loading size base on the source code.

When the ASP application is running, a welcome page and a finish page will show up, which is index.asp file (load size: 1k bytes) and end.asp file (load size: 1k bytes). These two files are just index and finish pages and do not count in the loading size.

The main initial load size of the ASP application and first question required is about 8k bytes overhead, including the main ASP application file (Mlearning.asp, and a header image file); the JavaScript file controls the download question, verifies the submitted form and submits the answer back to the server. The initial loaded file size includes the load size for the first question.

Table 5.3: ASP application initial load file sizes

File Name	Load Size(bytes)	Detail
mlearning.asp	2k × 2	Load first questions, and refresh the whole page after submitted the answer.
iq_header.gif	2k	Header image file
asp_m.js	2k	JavaScript file
hello.css	45	CSS file
Initial Load Size	8k bytes	

The header image file, JavaScript file and the CSS file need to be loaded only once if the browser supports caching. In our measurement, all browsers (mobile base or desktop base) support caching.

The mlearning.asp loaded size (2kb) is smaller than its file size (3kb) (Table 5.1). When a browser requests an HTML file, the server returns the file, the loaded size is the same as the file size. Because the ASP file executes to generate returnable plain HTML file, not all the codes will be sent back to the browser. So, mlearning.asp file load size (2kb) is the same as the mlearning.asp file size without the asp script, plus the bytes of the question.

Because of the nature of the ASP file, all the ASP script needs to be executed on the server side. Then, the main application file (mlearning.asp) needs to be loaded every time the user requests any new questions and refreshes the whole page after submitting the answer. First, the Asp file fetches questions from the ACCESS database, generating plain HTML questions file that returns back to the browser. Second, the answer needs to be posted back to the server, then verified by the server side ASP engine; then it sends the result (plain HTML, about 2kb) back to the browser. So, the client side Web browser will download $2k \times 2 = 4K$ bytes for each question.

During the process, the mobile learning application contains 10 multi-choice questions total, but when the application initially loads, only one question had been loaded. Then the mlearning.asp file only needs to load plain HTML nine times to generate the other nine questions for the user. Additionally, it sends back nine times after the answer is verified. So, the mlearning.asp file will load $2 \times 9 = 18$ times and generate about $2k \times 18 = 36k$ bytes of plain HTML transmission data. Total load size formula:

$$\text{Initial Load Size} + \text{Process Load Size} = \text{Total Size} \quad (1)$$

So, the total data load size for the whole ASP mobile learning Web application is:

$$8K + 36K = 44K \text{ bytes}$$

As a result, we expect about 44k bytes data need to be sent back to the browser from the M-learning Web server.

5.2.2 Theoretical Ajax system performance

When an Ajax application is requested by a browser, an HTML page is loaded into the browser. This HTML page contains the JavaScript needed to run the user interface and to issue XMLHttpRequest calls, the XML handler to format and present the results, and any other elements needed for the user interface, such as HTML code and Cascading style sheets (CSS). The client interacts with the user interface presented on the browser page (e.g. click hyperlink, submitting forms, or triggering an event). The Ajax engine handles all interaction events between user and server by generating an XMLHttpRequest message to the Web server. The Web server returns new content in XML file format. This XML data is handled by the Ajax engine and presented to the user by Web page partial update.

The same as the Asp version application, the Ajax M-Learning displays a welcome page at the beginning (index.htm), which is about 1K bytes and a finishing page at the end (end.htm). These two files are the index and finish page, and do not count in the loading size.

The main application file (mlearning.htm) and serial common elements will be loaded into the Web browser (e.g, JavaScript file, CSS file, and header image file) for the first question. So, this initial load of the Ajax application requires 10k overhead, but it is different from the ASP Web application, which is the XML file only needs to be downloaded once for each question.

Table 5.4: Ajax application initial load file sizes

File Name	Load Size(bytes)	Detail
2.asp	500	Send XML back to browser (XML file size)
majax.js	5k	JavaScript file(Ajax Engine)
hello.css	45	CSS file
iq_header.gif	2k	Header image file
mlearning.htm	2k	Load first questions
Initial Load Size	10k bytes	

The same as the ASP version, the header image file, JavaScript file and the CSS file need to be loaded only once if the browser supports caching.

In the Ajax version, the M-learning application, the initial load size is more than the ASP version application. But the advantages of the Ajax application will show up, when the user is using the application. No ASP file needs to be downloaded from the Web server, only the question's content and answers will be sent back to the browser in XML file format. All the answers from the user can be verified locally by the Ajax engine; the browser does not need to submit any data back to the server. The expectation of XML file sent back from the Web server is about 500 bytes, according to the structure of the XML file and its content (Figure 5.6). It means only about 500 bytes of data need to transfer between the server and browser. Then the Ajax engine can partially update the Web page to tell the user the correct answer locally by the XML file.

To illustrate, M-learning application contains 10 multi-choice questions. In the Asp version, a main application file (asp file) needs to load twice (first, load the question; second, load the answer) for each question, and needs to load the entire file including the Webpage layout (about 2000 bytes; the question is about 500 bytes, and the rest of the bytes are the Webpage formatted informant). But in the Ajax version, the Web server only sends back the question without any formatting (500 bytes). So, the main Ajax application only needs to transfer about $9 \times 500 = 4.5\text{k}$ bytes of data during the process.

The main program needs 4.5kb data transfer between the Web server and the client side. So, according to the algorithm (1), the total data load size for the whole Ajax mobile learning Web application is:

$$10\text{K} + 4.5\text{K} = 14.5\text{K bytes}$$

For the Ajax Web application, we expect about 14.5k bytes data needs to be sent back to the browser from the M-learning Web server.

In summary, according to the calculation, the Asp M-learning system needs to transfer 44k bytes of data; Ajax can provide the same interface by only transferring 14.5k bytes of data. That is a big improvement. Is it the same in the real Web server measurement? A real-life performance measurement will be performed.

5.3 Real-life performance (Vodafone New Zealand GPRS)

As discussed before, a traditional Web application uses a multi-page model; if Ajax uses single-page model, then the loaded data size should be different in real-life performance measurement. This section collects the experimental data from the server in the real-life environment, including bandwidth usage, and task implementation time. The experiment measured the performance of an Ajax enabled Website over the GPRS network.

5.3.1 Vodafone New Zealand GPRS network

The experimental test bed setup is very important to measure the M-learning system. It shown in figure 5.10, we used a laptop connected to an iTegno 3000 GPRS modem by USB 1.1 through a serial PPP (point-to-point) link to act as a GPRS mobile terminal to emulate the GPRS connection from mobile devices to the Web server in the Vodafone New Zealand (GPRS/GSM 900/1800/1900MHz) network.

iTegno 3000 is a GPRS modem that uses GSM 900/1800MHz, has embedded antenna and is powered by USB, and also supports extended GSM (EGSM) and GPRS. It can use more than one GRPS coding scheme (CS1, CS2, CS3, and CS4), has GPRS functionality, and Class 2 “2+1” multi-slots (2 downlink, 1 uplink channels).

The test laptop uses Windows XP operating system; the Web browser uses Opera Mobile 8.65 for Windows mobile 5/6 PPC in Windows mobile 6 emulator. The Vodafone New

Zealand GPRS network (two download channels and one upload channel, CS4) is being used. And the wireless network signal level may be difference in different location, to avoid the difference between different locations. All access to the Web server from the test laptops is using the Albany GSM Base Station, according to the IP address and time.

The Web server is running Internet Information services 6.0 (IIS 6.0); system logs have been active in the IIS, all traffic visits going to and from the Web server will be recorded into the system log. It enables us to isolate and monitor all the transfers between the Web server and mobile terminal. All collection data can help us analyse and measure the Ajax Web application performance over the GPRS network.

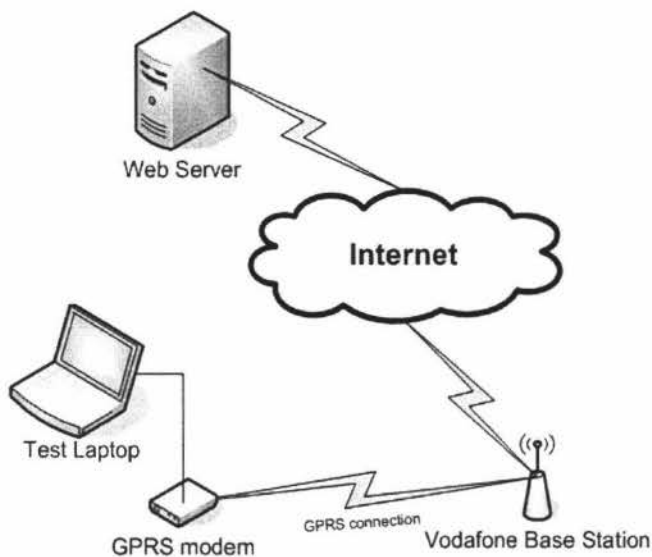


Figure 5.10: Real-life test bed

All data was collected by the system log, but the system log uses a sequential way to record all the traffic and data transmission. Log Parser is an IIS log analysis tool from Microsoft.

Log Parser 2.2 is available as a stand-alone executable (LogParser.exe) and a COM interface (logparser.dll) and both are available in the Log Parser installation. IIS uses W3C extended log format (Microsoft TechNet, 2007).

Over 200 hits were posed using the ASP version of M-learning system and also using the Ajax version of M-learning system. Collecting the Web server transmission data can help us analyse the performance for ASP M-Learning application and Ajax M-learning application.

Table 5.5 is an Internet Information Service (IIS) log example for the ASP mobile Learning Web application. It has “GET” and “POST” method.

Table 5.5: ASP Application Log Example

cs-method	cs-uri-stem	sc-bytes	cs-bytes	time-taken
GET	/asp1/index.asp	1117	466	468
GET	/asp1/hello.css	346	454	156
GET	/asp1/iq_header.gif	1815	459	296
GET	/asp1/asp_m.js	1893	469	187
GET	/asp1/index.asp	1117	407	140
GET	/asp1/iq_header.gif	1815	352	156
GET	/asp1/mlearning.asp	2181	493	265
POST	/asp1/mlearning.asp	2274	686	499
	.			
	.			
	.			
GET	/asp1/mlearning.asp	2184	495	203
POST	/asp1/mlearning.asp	2280	690	343
GET	/asp1/end.asp	963	460	156

Table 5.6 Internet Information Service (IIS) log example for the Ajax mobile learning Web application. Only one kind of action method Ajax application has which is “GET” method.

Table 5.6 Ajax application log example

cs-method	cs-uri-stem	sc-bytes	cs-bytes	time-taken
GET	/ajax1/index.htm	851	508	171
GET	/ajax1/hello.css	346	350	140
GET	/ajax1/iq_header.gif	1814	354	281
GET	/ajax1/mlearning.htm	2166	515	374
GET	/ajax1/majax.js	5190	353	296
GET	/ajax1/2.asp	518	377	296
GET	/ajax1/2.asp	514	377	203
GET	/ajax1/2.asp	500	376	187
GET	/ajax1/2.asp	621	377	187
GET	/ajax1/2.asp	659	377	187
GET	/ajax1/2.asp	528	376	234
GET	/ajax1/2.asp	533	377	203
GET	/ajax1/2.asp	571	378	249
GET	/ajax1/2.asp	583	377	203
GET	/ajax1/2.asp	524	379	187
GET	/ajax1/end.htm	1951	465	156

These log examples were collected from the real-life test Website: <http://www.ajaxnz.com>

5.3.2 Real-life performance — Response Size

Response size means the number of bytes that the Web server sent that it is one of the most useful measurable features for application performance. In the IIS log, its symbol is

“sc-byte”. On the other hand, the Web application performance can be improved by small amounts of transmission and transfer times. Collecting the server transfer details and times can help us understand the application’s usability.

ASP M-Learning application

Table 5.5 shows all details from the Web server log for the ASP Web application. “cs-method” means the requested action. “se-bytes” means the number of bytes that the server sent. “cs-bytes” means the number of bytes that the server received. “time-taken” means the length of time that the action took in milliseconds (Microsoft TechNet, 2007). It shows one of the most important facts for the ASP Web application which is the “mlearning.asp” file needs to be loaded two times for each question; the first time is the “get” request for the question, and the second time is to “post” request for the answer, which refreshed the whole page.

Table 5.7: ASP application “sc-bytes” statistic

cs-uri-stem	sc-bytes		
	AVG	MIN	MAX
ASP application			
/asp1/mlearning.asp	2305 × 2	2256	2406
/asp1/asp_m.js	1893	1833	1897
/asp1/iq_header.gif	1815	1800	1880
/asp1/hello.css	346	356	396
Initial load size.	≈ 8.7Kb		

According to the IIS system log (Table 5.7). ASP M-Learning application initial load average size is about 8.7Kb (including the first question). As expected, the main ASP application file needs to load 10 times to display the questions and load another 10 times

after users submit the answer. “Mlearning.asp” file average loading size is 2305Kb, following the methodology (1); the total bytes need to be sent back from the Web server:

$$8.7\text{Kb} + 9 \times 2 \times 2305 \approx 50\text{Kb}$$

So, in real-life performance tests, the ASP M-learning application needs to load about 50Kb of data from the Web server during the whole application process.

Ajax M-Learning application

The number of bytes returned for Ajax application is different from the ASP application. First, the load size is very different from the ASP version (Table 5.7). It needs about 10Kb of initial load data, the major file is the JavaScript (Ajax engine) file which is about 5Kb, and the other one is an HTML layout file that is about 2Kb (Table 5.8).

Table 5.8 Ajax application “sc-bytes” statistics

cs-uri-stem	sc-bytes		
	AVG	MIN	MAX
Ajax application			
/ajax1/majax.js	5190	5100	5280
/ajax1/mlearning.htm	2166	2007	2356
/ajax1/iq_header.gif	1814	1754	2014
/ajax1/hello.css	346	326	386
/ajax1/2.asp	555	500	614
Initial load size.	≈ 10kb		

But during the Ajax M-learning run, just a few of bytes needed to be transferred between the client browser and Web server (Table 5.6). All this data is XML files and the average

sent back size is about 555 bytes only (2.asp file), including the question and the answer. It allows the Ajax engine to verify the answer locally, and provides users very fast response time. Also, because the question and answer loads in at one time, then each question only needs one post back from the Web server. According to the total load size formula (1):

$$10\text{Kb} + 9 \text{ times} \times 0.555\text{Kb} \approx 15\text{Kb}$$

So, it only needs to transfer about 15Kb of data between the Web server and client browser for the real-life Ajax M-learning application.

Compared with the server sending out data for the ASP Web application and Ajax Web application, there is huge difference. For ASP Web applications, it needs the server to transfer 50K bytes of data during the whole Web application, and only 15K bytes of data from the server is sent back to the client side for the Ajax Web application.

5.3.3 Real-life performance — Response time

Besides the transfer data, the server's response time (or time taken) is another important performance feature that can measure the Web application's performance. In the IIS system log, there is a field called "time-taken", which is the length of time that the action took in milliseconds on the server, or can be considered as the server's action response time.

ASP M-Learning application

The measurement formula is similar to measuring the bytes sent back from the server. According to the IIS system log (Table 5.9), the initial response time for Asp M-learning application is about 1142ms (initial load time including the first question loading time), and the main application file needs to load double the number of times as the questions.

The average response time for “mlearning.asp” file is 285ms; the “mlearning.asp” needs to process twice for each question, then for a total of 10 questions, minus the response time for first question, 9 questions are left. So the process response time is $9 \times 2 \times 285 = 5130\text{ms}$.

Table 5.9 ASP application “time-taken” statistics

cs-uri-stem	Time-taken		
	AVG(ms)	MIN(ms)	MAX(ms)
ASP Application			
/asp1/mlearning.asp	285×2	187	499
/asp1/iq_header.gif	207	156	296
/asp1/asp_m.js	202	187	218
/asp1/hello.css	163	156	171
Initial Time taken(ms)	1142		

Similar to the total load size formula, the total response time for the whole application formula is:

$$\text{Initial Response Time} + \text{Process Response Time} = \text{Total Response} \quad (2)$$

So, the whole Web server response time for the whole ASP mobile learning Web application is:

$$1142\text{ms} + 5130\text{ms} = 6272\text{ms}$$

The total time taken on the Web server is 6272ms for the whole ASP m-learning application.

Ajax M-Learning application

For the Ajax M-Learning application, the IIS system log shows the initial response time is about 1378ms (including the first question), and the average server response time for

the XML generator “2.asp” file is 210ms (Table 5.10). But “2.asp” only runs one time for each question (Table 5.6) on the server. Then for a total of 10 questions, minus the response time for the first question, 9 questions are left. So the process response time is $9 \times 210 = 1890\text{ms}$.

Table 5.10 Ajax application “time-taken” statistics

cs-uri-stem	Time-taken		
	AVG(ms)	MIN(ms)	MAX(ms)
Ajax Application			
/ajax1/hello.css	319	140	499
/ajax1/majax.js	296	296	296
/ajax1/iq_header.gif	281	201	381
/ajax1/mlearning.htm	272	171	374
/ajax1/2.asp	210	171	343
Initial Time taken(ms)	1378		

According to the formula (2), the whole Web server response time for the whole ASP mobile learning Web application is:

$$1378\text{ms} + 1890\text{ms} = 3268\text{ms}$$

The total time taken on the server side is 3268ms for the whole asp m-learning application. In short, for the whole Mobile Learning application, the ASP application needs about 6272ms, but the Ajax application needs only 3268ms.

5.3.4 Real-life performance — Query size

The bytes response from Web server and the task time taken by the Web server show us the Ajax M-learning application transfers small amounts of bytes between the client and server and it has a better response time than ASP M-learning application. What about the

data submitted from the client? Does it show us the same of the figures or does it have different results?

The query size can affect the transfer time; if the query size is big, not only does it need more bandwidth to transfer the data, but also it takes a long time to transfer before the server can actually process the request. Then from the user point of view, this Web application response time is slow. Then, evaluating the client Web browser query size is necessary.

Field “cs-bytes” in IIS system log shows us the number of bytes that the server received. So, this data can be considered as the data sent out from client browser to request new data from server. The log records each query size in “cs-bytes” field.

ASP M-Learning application

Table 5.11 shows query size of submitted data from the client to the Web server. For the “mlearning.asp” file there are “GET” and “POST” method action, the query size is different between these two methods.

Table 5.11 ASP application “cs-bytes” statistics

cs-uri-stem	cs-bytes(bytes)		
	AVG	MIN	MAX
/asp1/mlearning.asp(“GET”)	496	492	566
/asp1/mlearning.asp(“POST”)	686	600	750
/asp1/iq_header.gif	398	352	459
/asp1/asp_m.js	423	378	469
/asp1/hello.css	416	379	454
Initial Query Size	2428 (including first question)		

These queries are used to request the next question for the user. The average query size for ASP M-learning main file “mlearning.asp” is about 496 bytes (Table 5.11, “GET” method). Of course, in the ASP version, users need to submit his/her answers back to the Web server to verify the answer. The average transfer bytes are 686 bytes (“POST” method) when users submit answers. So, the initial query size (including first question) formula is:

$$\text{mlearning.asp (GET) + mlearning.asp (POST) + other file} = \text{Initial Size} \quad (3)$$

According the formula (3), the initial query size is:

$$496+686+398+423+416=2428 \text{ bytes}$$

And the total query size calculation formula is:

$$\text{Initial query size + process query size} = \text{Total Query Size} \quad (4)$$

Then,

$$2428 + (686 \times 9 + 496 \times 9) = 13066 \text{ bytes} \approx 13\text{Kb}$$

When users finish the whole M-Learning Web application, the client side Web browser needs to submit about 13Kb data to the Web server.

Ajax M-Learning application

Table 5.12 shows the Ajax Web application query size or submit data from client to the Web server. And as Table 5.6 shows only one method action will occurs during the whole application process, “GET” method.

Table 5.12 Ajax application “cs-bytes” statistics

cs-uri-stem	cs-bytes		
	AVG (bytes)	MIN (bytes)	MAX (bytes)
Ajax Application			
/ajax1/2.asp	377	373	379

/ajax1/iq_header.gif	354	350	374
/ajax1/majax.js	353	351	363
/ajax1/hello.css	350	350	370
/ajax1/mlearning.htm	515	514	519
Initial Query Size	1949 (including first question)		

The average query size is 377 bytes in the Ajax M-Learning Web application. These submissions only request the next question: one question, one submission. Then, to calculate the initial query size (including the first question) for the Ajax application is very easy; add all query sizes together:

$$377+354+353+350+515=1949 \text{ bytes}$$

For 10 question application, according to the formula (4), the total query size is:

$$1949+9 \times 377=5342 \text{ bytes} \approx 5.3\text{Kb}$$

So, when users finish the whole M-Learning Web application, the client side of the Web browser needs to submit about 5.3Kb data to the Web server.

The result is that the ASP Web application submitted around 13Kb of data back to the server; while the Ajax Web application sent back about 5.3Kb of data. The query size is collected from the test laptops as described before. But the query size may have wide variations due to the different test platforms or different Web browsers.

5.4 Usability evaluation

5.4.1 Usability test overview

The goal of this project is to evaluate the Ajax enabled mobile learning system's performance over the GPRS network. The performance will be evaluated by the data statistics and user experience. The user experience evaluation relied on good usability testing. So, the goal of this real-life (real user, real Website, real mobile device and real GPRS network) usability test is to find out: Does user notice difference between the Ajax and the non-Ajax Web-based mobile learning application? Is it easy to notice or hard to notice the differences? Where Web-based mobile learning application they prefer to use and why?

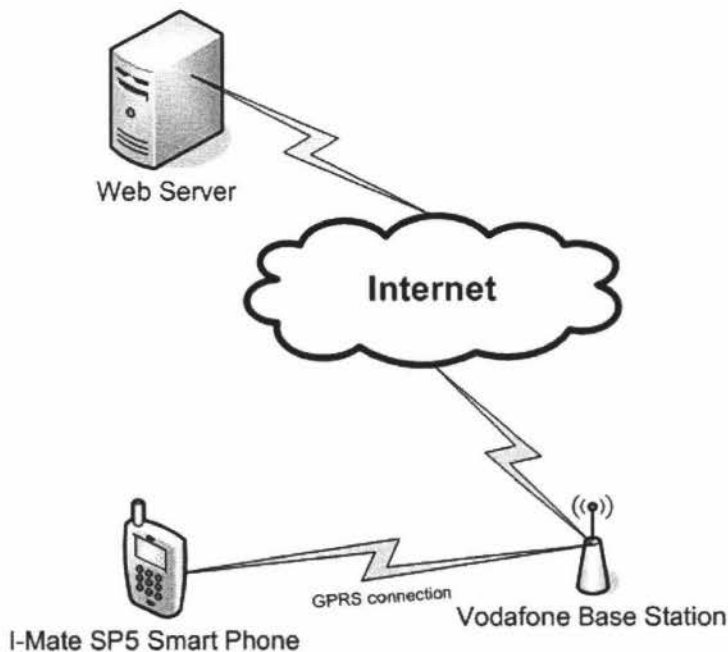


Figure 5.11: Usability testing bed

In order to do the best to simulate the real-life environment, the usability testing test bed is a little bit different from the performance measurement test bed in section 5.3. The test bed shown as Figure 5.11, we used the I-Mate SP5 Smart-Phone instead of the laptop. The mobile Web browser is using Opera Mobile, version 8.65 for Windows mobile 5/6

Smart-phone operating system. The same Vodafone New Zealand GPRS network (2 download channels and 1 upload channel, Coding Scheme 4) was used. All users are using the same I-Mate SP5 Smart-Phone under the Albany GSM Base Station. During the whole test, make sure the GSM signal level is strong.

There are two Mobile Learning Quiz Web application ready (Ajax and ASP) in the Internet that has the same layout, user interface and questions. Users need to finish them via mobile device (I-mate SP5 smart phone), think aloud about their feelings and answer the questionnaire based on the feeling. In order to reduce the interview time, we changed the mobile learning questions from 10 questions down to 5 questions for each system.

There are two tasks in this usability test. Task one require users to finish all multi-choice test questions in Ajax mobile learning system by I-mate SP5 Smart Phone. Task two requires users to finish all multi-choice test questions in ASP mobile learning system by I-mate SP5 Smart Phone. A training section will provide before user start their mobile learning activity.

In this usability test, there are two variables we collected from the users to represent both quantitative data and qualitative data. The first quantitative variable is the task completion time which it uses to measure two different performances. It is within-subject design. The second variable is a qualitative variable, the preferred data, which is used to find out if users notice the difference between the two Websites.

5.4.2 Usability test

This is anonymous voluntary usability testing; the whole usability testing process takes about 5~10 minutes (Appendix 4) for each user and they can stop it any time they want. There are three steps to finish the usability test.

First, the user will be invited to the testing (Appendix 3). If they agree, then, they have some time to get familiar with the test device (I-Mate SP5), a short tutorial and demo will apply. During the tutorial, we will introduce the I-Mate SP5 smart phone, teach them how to use the smart phone to access the Website by Opera Mobile, and tell them the usability test process. The demo system is exactly the same as the ASP mobile learning system, only questions are different.

We will outline all questions (Appendix 7) the mobile learning, give user some background about the mobile learning system, and tell all users that we will not collect any answer for question, they can select any answer they want; may be the answers are all wrong, it would not affect our usability test. This is an important part of the usability test, because user may concentrate on the question display on the mobile phone, instead of the whole mobile learning system.

Second, when the participant finished the demo system, which can help user to get familiar with the Opera Mobile and the I-Mate SP5 phone, the real usability test will start. Users can talk or ask any questions they want during the test, we will provide any necessary support to help user finish the test and record down all useful information for further study, including how long the participant takes to finish for each mobile learning system.

Finally, the “post-walkthrough” method will apply to wrap up the test; we will finish the questionnaire (Appendix 5) and clean the Opera Mobile cache for the next tester.

The questionnaire includes some user information, such as, “Gender”, “What kind of mobile phone do you have?” “Have you ever used your mobile phone to access the

Internet, if so, how often?” And the feeling for each mobile learning system, “Did you notice any differences between these two Website?” “Which one you prefer to user?”

CHAPTER 6: Results and discussion

This chapter will discuss and analyse the collected data in chapter 5, and aims to find out the performance difference between ASP and Ajax Web applications.

6.1 Data analysis

The Ajax web application and ASP web application have much in common, they showed quite different performance in our experiment. The data collected from the IIS web server system log comprised hundreds of entries, so all results stated here are averages.

First, the initial load size of the applications has some impact on performance. When the client-side Web browser loads the Web application for the first time, some files need to be downloaded. The ASP Web application's initial load size is about 8.7Kb, including an ASP file, a JavaScript file, an image file, and a CSS file. It takes the server about 1142ms to respond. For the Ajax Web application, the initial load size is about 10Kb, with a 1378ms server response time. In this case an HTML file, a JavaScript file, an image file, and a CSS file will be loaded. The image and the CSS file are the same, and there is a small difference between the sizes of the HTML and ASP file. The major difference, however, is in the JavaScript, because the Ajax JavaScript file is about 50% larger than in the ASP version. This is because the Ajax JavaScript contains some additional logic and layout control functions.

The second performance difference is evident when the Web application is running. For each multi-choice IQ test question, the ASP version needs to fetch the question page first and then refresh the page again with the response, after the user has submitted their answer. This requires two download activities from the server. The Ajax version only

requires one download activity for each question, because the JavaScript can verify the user's answer on the client, avoiding the second request-response cycle. The ASP version cannot do this because it can only run on the server.

There are nine questions in the test application, after the initial question. For the ASP version, each question needs to transfer 4.6Kb of data, with a 570ms server response time. Therefore, nine questions need to transfer 41.4Kb of data and use 5,130ms of server time. For Ajax, an average of only 555 bytes of data needs to be transferred for each question and the server takes 210ms to respond. Therefore, nine questions need 4.995Kb of data and 1,890ms of server time.

For the query size, the ASP application's average initial query size is 2428 bytes. And for each question, 496 bytes query size to fetch the question and 686 bytes query size to submit the user's answers. So, the subsequent query size is about 11Kb in ASP application. Compare with one query for each question in Ajax application, the query is not small. The initial query size is about 2Kb, and there is about 4Kb subsequent query size.

Taking both initial load and subsequent page loads into account, the ASP application needs to send out 13Kb of data and transfer a total of 50 Kb of data compared to only 5.3Kb of data need to be sent out and transfer 15Kb for the Ajax application, and takes a total of 6,272ms to respond compared to only 3,268ms (Table 6.1).

In "Measuring the Benefit of Ajax" and "Modelling Ajax Application Performance", White (2005), Smullen and Smullen (2006) use the following algorithm to define the percentage of Ajax's application performance improvement:

$$\frac{(\text{HTML-Ajax})}{\text{HTML}} \quad (5)$$

The average data bytes sent from Web server is 50 Kb for ASP version and 15 Kb for the Ajax version during the whole in the M-Learning Web application. The average data bytes retrieved by the Web server is about 13Kb for the ASP version and about 5.3Kb for the Ajax version during the entries in the M-Learning Web application. It takes about 6272ms for Web server to finish all queries for the ASP version; it only takes about 3268ms for the Ajax version.

Table 6.1: Performance improvement

	Response bytes	Response seconds	Query size
ASP	50Kb	6272ms	13Kb
Ajax	15 Kb	3268ms	5.3Kb
Improvement	71.00%	48.00%	59.23%

A good design and development Web application can provide users the best Web application experience. In the traditional Web application, the user's waiting time can be separated into three parts: data transfer time (two ways, client to server and server back to client) and server process time. The measurable benefits are shown on Table 6.1; the Ajax approach in our experiment provides almost a 50% performance improvement when using a mobile device over GPRS. Ajax M-learning Web application has much better performance than the non-Ajax M-learning Web application in a GRPS mobile phone network.

6.2 Usability test results

Twenty people attended this small group usability testing, totalling 19 valid examples; one participant didn't finish the test. In all of the valid examples, 18 of them have mobile phones, such as, Sony, Nokia, and Motorola, and so on; only one participant doesn't have mobile phone. There were 6 female participants and 13 male participants. 42.1% of participants had experienced using the mobile phone to access the Internet before. 20%

had accessed the Internet a few times, 20% have used the mobile Internet just once. Only 10% of the users used mobile Internet very often, once a week (Appendix 6).

During the usability test, because of the demo system, all participants are comfortable about the mobile learning system itself, but there still have some problems. The first complain is about the navigation button, it is hard to use and not functioning very well. Sometime users have to keep pressing it until it works (action on the screen). One participant aborted the test, because he is not good at the navigation button. The second problem is about the small screen and front size is small. In fact the front size and the Webpage size can be adjusted. Using the same screen, display question and answer in one screen, for all users can help us to reduce the user interface ingredient. Clean the mobile web browser's cache to each participant force the system using the latest update of the learning system.

In fact, because of the asynchronous connection model and partial loaded page update by Ajax engine, the Ajax mobile learning system can display the question before the question's options. During the usability test, one participant noticed this fact, he is a programmer and knows Ajax approach, other participants didn't notice that the question and option update in different code statement.

Based on the 19 test examples, applying pair-wise T-Testing to the task completion time for examples, it shows in terms of time, our participants significantly better performed with the Ajax mobile learning system.

$$t(18)=11.71, \quad p<0.05.$$

The means value of the Website one task completion time is 119.26 seconds and the Website two mean task completion time is 64.11 seconds. It means users finished task

two on Website two faster than users' task completion time on Website one, about 50% faster. Ajax mobile learning system had much better performance than ASP mobile learning system. This is the measurement on the quantitative variable.

In terms of frequency to their preference, we can use chi-square tests, but to some extent it does not seem to meet its assumption.

$$\text{Chi-square (2)} = 12.41, p < 0.05$$

The expected sample is too small, so it would be unwise to use this statistical analysis for sure. However, analytically (not statistically) speaking, that users noticed there were some differences between the two Websites, even though they have the same interface, the same learning process. There are 12 out of 13 participants, who noticed the difference preferred the Ajax mobile learning system.

These two results from the usability test indicate that Ajax mobile learning system can provide better performance in the mobile environment on the user experience level.

To sum up, we apply Ajax approach to a mobile learning system and compare it with the traditional ASP Web-based mobile learning system. After the evaluation of these mobile learning systems from the performance measurement and the usability test, the result is that Ajax Web-based mobile learning system can reduce the transmission volume, server's response time, and has more users prefer to use. Moreover, applying Ajax approach to all the Web-based mobile application should be the easy way to improve the system performance than updating the mobile hardware or the wireless network.

CHAPTER 7: Conclusion

Ancient Chinese saying: "All good things come to an end".

The last part of this report provides a conclusion for this Ajax approach performance research on the mobile phone environment. The recommendations from this report and limitations of this study are outlined after the conclusion. The final section of this chapter gives some suggestions for future studies. All suggestions correspond with the limitations of this study.

7.1 Summary conclusion

Applying Ajax approach in a mobile learning system can increase the usability of mobile learning system. The advantages of using Ajax in mobile learning system are, first, it allow user to access to the system as long as his/her mobile browser support Ajax and have GSM/GPRS network, and it allows mobile learner study anywhere anytime. Second, the mobile learning system performance is much faster than the traditional HTML base mobile learning system. Third, the asynchronous and text-based transmission can reduce the Web server response time and response bytes, it can save money for mobile learners (most of mobile telecommunications companies charge for the amount of data sent and received, e.g. Vodafone New Zealand). Finally, the mobile learning system also can be accessed from desktop environment.

Technically, Ajax reduces 71% network transfer traffic from the server to the client, saving 48% of the Web server action processing time and decreasing 59.23% of the submission data from the client to the server. Furthermore, on the user experience level, the task completion time statistics show users can finish the same M-learning

multi-Choice questions on the Ajax system 50% faster than they are working on ASP M-learning system over the GPRS network. More than 60% (12 out of 19) users noticed the difference between these systems and prefer to use the faster one (Ajax M-learning system).

The results of our experiment demonstrate that the Ajax approach can significantly reduce both the data transmission size and the server's response time. Meanwhile, reducing the bandwidth required, and speeding up the user interface on the mobile device provides the user with a better mobile Internet experience. In addition, Ajax approach can save developers' time and money, because it is a software technology and does not require any hardware or environment update to implement.

7.2 Recommendations

The Ajax approach can significantly reduce the bandwidth usage, server's processing time and the network transfer time to provide users with a good mobile Internet user experience. The main recommendation for this research is that Ajax is very suitable on mobile Web application system. It allows developers to setup only one Website to provide good user experience on PCs and mobile devices and provide users some enjoyable mobile learning experiences.

7.3 Limitations of the study

The limitations of the research are outlined as following:

- Only tested in one GPRS network, 2 download channels and 1 upload channel, and only one mobile device (I-mate SP5). There are some other GPRS network configuration, such as, 3 download channels and 1 upload channel, or 2 download channels and 2 upload channels, can be used as an evaluation network.

- The Web application functionality is too limited, only measuring the Ajax performance in multi-Choice Mobile Learning Website. Our mobile learning system also includes Glossary, Dictionary and so on. These functions are not a very good for an Ajax performance sample.
- The usability test group is too small and narrow (most of the participants are students or teachers).

7.4 Suggestions for future research

Some suggestions for the future research relate to this research. First, in this research, both ASP and Ajax Web applications have the same user interface to avoid the user interface infection. But in some cases, the Ajax Web application has better performance than the ASP Web application, if they are using different kinds of UI. The Ajax approach uses the Ajax engine, the page update and asynchronous network connection.

Second, test these Web applications in different kinds of GPRS networks or different connection settings for mobile devices, such as three download channels, and one upload channel. It can find out the network feature infection for the Ajax approach, or researchers can try another kind of Web application beside the multi-choice.

Finally, the last suggestion is to develop the Ajax Web browser on mobile devices to allow all Websites to update partially even if it is not an Ajax enabled Website.

References

- ajaxpatterns.org. (2007). Java Ajax Frameworks. Retrieved 24th Oct 2007, from http://ajaxpatterns.org/Java_Ajax_Frameworks
- ajaxtags.sourceforge.net. (2007). Ajax JSP Tag Library 1.3. Retrieved 24th Oct, 2007, from <http://ajaxtags.sourceforge.net/>
- Akkawi, A., Schaller, S., Wellnitz, O., & Wolf, L. (2005). Network Mobile Gaming for 3G-Networks. Retrieved 20th Oct 2007, from <http://www.ibr.cs.tu-bs.de/users/wellnitz/papers/misc/ims-icec2004.pdf>
- Alexander, B. (2004). Going Nomadic: Mobile Learning in Higher Education. *EDUCAUSE Review*, 39(5), 28-35.
- Angelaccio, M., & Buttarazzi, B. (2006). *A Performance Evaluation of Asynchronous Web Interfaces for Collaborative Web Services*. Paper presented at the Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops.
- Attewell, J., & Savill-Smith, C. (2003). *Learning with Mobile Devices: Research and development. MLEARN '03 Book of Papers*. London: Learning and Skills Development Agency.
- BBC.com. (2006). Big growth in mobile phone sales. Retrieved 24th May 2007, from <http://news.bbc.co.uk/1/hi/business/4459278.stm>
- Berri, J., Benlamri, R., & Atif, Y. (2006). *Ontology-based framework for context-aware mobile learning*. Paper presented at the Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada.
- Black, J. T., & Hawkes, L. W. (2006). *A prototype interface for collaborative mobile learning*. Paper presented at the Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada.

- Brasche, G., & Walke, B. (1997). Concepts, services and protocols of the New GSM Phase 2+ General Packet Radio Service. *IEEE Communications Magazine*.
- Brown, R., Ryu, H., & Parsons, D. (2006). *Mobile helper for university students: a design for a mobile learning environment*. Paper presented at the Proceedings of the 20th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artefacts and environments, Sydney, Australia.
- Cao, Y., Tin, T., McGreal, R., Ally, M., & Coffey, S. (2006). *The Athabasca University mobile library project: increasing the boundaries of anytime and anywhere learning for students*. Paper presented at the Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada.
- Caudill, J. G. (2007). The Growth of m-Learning and the Growth of Mobile Computing: Parallel developments. *The International Review of Research in Open and Distance Learning*, 8(2), 1492-3831.
- Cerise, W., Gail, K., Terry, G., & Nelson, P. (2003). On-line instructional testing in a mobile environment. *Journal of Computing Sciences in Colleges* 18(4), 23-29.
- Chakravorty, R., Cartwright, J., & Pratt, I. (2002). *Practical experience with TCP over GPRS*. Paper presented at the Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE.
- Chakravorty, R., Clark, A., & Pratt, I. (2003). *GPRSWeb: Optimizing the Web for GPRS Links*. Paper presented at the MobiSys 2003: The First International Conference on Mobile System, Applications, and Services, San Francisco, CA, USA.
- Chakravorty, R., & Pratt, I. (2002). *WWW performance over GPRS*. Paper presented at the Mobile and Wireless Communications Network, 2002. 4th International Workshop Stockholm, Sweden.
- Che, X., Hamalainen, S., Ryyanen, J., & Moisio, M. (2003). *GPRS radio network performance simulation and optimization with dynamic simulator*. Paper presented at the Communication Technology Proceedings, 2003. ICCT 2003. International Conference on.

Church, K., Smyth, B., Cotter, P., & Bradley, K. (2007). Mobile information access: A study of emerging search behavior on the mobile Internet. *ACM Trans. Web, I(1)*, 4.

CNN.com. CNN Timestamped: 22nd April, Updated: 0910GMT.

Crane, D., Pascarello, E., & James, D. (2005). *Ajax in Action*: Manning Publications Co.

Dix, A. J., Finlay, J. E., Abowd, G. D., & Beale, R. (1998). *Human-Computer Interaction (Second Edition)*. Hertfordshire: Prentice Hall Europe.

Ergosoft Laboratories. (2003). What is a within-subjects design? Retrieved 23th Oct, 2007, from http://www.ergolabs.com/within_subjects.htm

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., et al. (1997). Hypertext Transfer Protocol -- HTTP/1.1. *RFC 2068*.

Garrett, J. J. (2005). Ajax: A New Approach to Web Applications. Retrieved July 12th, 2006, from <http://adaptivepath.com/publications/essays/archives/000385.php>

Gartner. (2006). Gartner Says Top Six Vendors Drive Worldwide Mobile Phone Sales to 21 Percent Growth in 2005. Retrieved June 3, 2007, from http://www.gartner.com/press_releases/asset_145891_11.html

Gartner. (2007). Gartner Says Worldwide Mobile Phone Sales Grew 21 Percent in 2006 Retrieved June 7th, 2007, from <http://www.gartner.com/it/page.jsp?id=501734>

Goth, G., & Costlow, T. (2007). Currents - The Google Web Toolkit Shines a Light on Ajax Frameworks. *Software, IEEE, 24(2)*, 94-98.

GSM Association. (2003). GPRS Roaming Guidelines. Retrieved August 29th, 2006, from <http://www.gsmworld.com/technology/gprs/index.shtml>

GSM Association. (2007). What is GSM? Retrieved 6th August, 2007, from <http://www.gsmworld.com/technology/what.shtml>

- Hanakawa, N., & Ikemiya, N. (2006). *A Web Browser for Ajax Approach with Asynchronous Communication Model*. Paper presented at the Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference.
- Henning, B., Gina, H., ussge, Guido, R., & ling. (2007). *An integrated system for interaction support in lectures*. Paper presented at the Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education, Dundee, Scotland.
- Hunaiti, Z., Garaj, V., Balachandran, W., & Cecelja, F. (2005). *An Assessment of 3G Link in a Navigation System for Visually Impaired Pedestrians*. Paper presented at the Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Proceedings. 15th International Conference.
- International Engineering Consortium. (2007). Wireless Application Protocol (WAP). Retrieved 25 August 2007, from <http://www.iec.org/online/tutorials/wap/>
- Internet Usage Statistics. (2007). Internet Usage Statistics - The Big Picture:World Internet Users and Population Stats Retrieved March 20, 2007, from <http://www.internetworldstats.com/stats.htm>
- JISC. (2005). *Innovative Practice with e-Learning* Bristol, UK: Joint Information Services Committe.
- Kadirire, J. (2007). Instant Messaging for Creating Interactive and Collaborative m-Learning Environments. *The International Review of Research in Open and Distance Learning*, 8(2).
- Kukka, H., & Ojala, T. (2006). *mobileDOK: culture in your pocket*. Paper presented at the Proceedings of the 3rd international conference on Mobile technology, applications and systems., Bangkok, Thailand.
- Kumar, V., Parimi, S., & Agrawal, D. P. (2003). WAP: present and future. *Pervasive Computing, IEEE*, 2(1), 79-83.
- Lee, T. B. (1992). What's new in '92. Retrieved March 7th, 2007, from <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/News/9201.html>

- Lee, W.-P., & Lu, C.-C. (2003). Customising WAP-based information services on mobile networks. *Personal Ubiquitous Comput.*, 7(6), 321-330.
- Lerner, R. M. (2006a). At the forge: Ajax application design. *Linux J.*, 2006(152), 9.
- Lerner, R. M. (2006b). At the forge: Beginning Ajax. *Linux J.*, 2006(151), 10.
- Lerner, R. M. (2006c). At the forge: Javascript. *Linux J.*, 2006(149), 10.
- Lerner, R. M. (2006d). At the forge: JavaScript, Forms and Ajax. *Linux J.*, 2006(150), 8.
- Lerner, R. M. (2007a). At the forge: Dojo Events and Ajax. *Linux J.*, 2007(156), 11.
- Lerner, R. M. (2007b). At the forge: Prototype. *Linux J.*, 2007(153), 11.
- Leung, C.-H., & Chan, Y.-Y. (2003). *Mobile learning: a new paradigm in electronic learning*. Paper presented at the Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference on.
- Lewis, C., & Rieman, J. (1993). *Task-Centered User Interface Design: A Practical Introduction*.
- Loken, S. B. (2000). WAP dead in 6 months? Retrieved 27th August, 2007, from <http://www.mail-archive.com/itb@itb.ac.id/msg18002.html>
- March, S., & Smith, G. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems* 15(1995), 251-266.
- Masinter, L. (1998). Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). *Internet informational RFC 2324*.
- Mesbah, A., & Van Deursen, A. (2007a). *An Architectural Style for Ajax*. Paper presented at the Sixth Working IEEE/IFIP Conference on Software Architecture (WICSA'07).

- Mesbah, A., & Van Deursen, A. (2007b). *Migrating Multi-page Web Applications to Single-page AJAX Interfaces*. Paper presented at the Software Maintenance and Reengineering, 2007. CSMR '07. 11th European Conference, Amsterdam.
- Microsoft TechNet. (2007). W3C Extended Log File Format (IIS 6.0). Retrieved 2nd August, 2007, from <http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/bea506fd-38bc-4850-a4fb-e3a0379d321f.msp?mfr=true>
- Mozilla. (2007). Mozilla web browser: Mozilla
- Nakahara, J., Hisamatsu, S., Yaegashi, K., & Yamauchi, Y. (2005). *iTree: does the mobile phone encourage learners to be more involved in collaborative learning?* Paper presented at the Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!, Taipei, Taiwan.
- Ni, S., & Haggman, S. G. (1999). *GPRS performance estimation in GSM circuit switched services and GPRS shared resource systems*. Paper presented at the Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE.
- Nokia. (2005). Nokia introduces a new Web browser for S60 3rd Edition. Retrieved 2nd November, 2006, from http://press.nokia.com/PR/200511/1019239_5.html
- Nortel. (2007). Long-Term Evolution (LTE): The vision beyond 3G. Retrieved 25th August, 2007, from <http://www.nortel.com/solutions/wireless/collateral/nn114882.pdf>
- Owen, C. L. (1997). Design Research: Building the Knowledge Base. *Journal of the Japanese Society for the Science of Design* 5(2), 36-45.
- Peters, K. (2007). m-Learning: Positioning educators for a mobile, connected future. *The International Review of Research in Open and Distance Learning*, 8(2).
- Pierce, C. S. (1931). *Collected Papers of Charles Sanders Pierce* (Vol. I-IV). Cambridge, MA: Harvard University Press.

- Pinkwart, N., Hoppe, H. U., Milrad, M., & Perez, J. (2003). Educational scenarios for the cooperative use of Personal Digital Assistants. *Journal of Computer Assisted Learning*, 19(3), 383-391.
- Polsani, P. (2003). *Network learning*. Vienna: Passagen Verlag.
- Purao, S. (2002). *Design Research in the Technology of Information Systems: Truth or Dare*. Paper presented at the GSU Department of CIS Working Paper, Atlanta.
- Rahul, V. T. (2001). Enhancing an e-commerce course with wireless application protocol (WAP) programming. *Journal of Computing Sciences in Colleges* 17(2), 5-13.
- Rosenberg, G. (2007). A look into the interaction design of the new Yahoo! mail and the pros and cons of AJAX. *interactions*, 14(2), 33-34.
- Rossi, M., & Sein, M. (2003). Design Research Workshop: A Proactive Research Approach. *Presentation delivered at IRIS 26*.
- Ryan, J. F., & Reid, B. L. (2006). *Usable Encryption Enabled by AJAX*. Paper presented at the Networking and Services, 2006. ICNS '06. International conference, Silicon Valley, CA, .
- Sayar, A., Pierce, M., & Fox, G. (2006). *Integrating AJAX Approach into GIS Visualization Web Services*. Paper presented at the Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference.
- Seong, D. S. K. (2006). *Usability guidelines for designing mobile learning portals*. Paper presented at the Proceedings of the 3rd international conference on Mobile technology, applications & systems, Bangkok, Thailand.
- Sharples, M. (2005). *Learning As Conversation: Transforming Education in the Mobile Age*. Paper presented at the Proceedings of Conference on Seeing, Understanding, Learning in the Mobile Age, Budapest, Hungary.
- Sharples, M., Corlett, D., & Westmancott, O. (2002). The Design and Implementation of a Mobile Learning Resource. *Personal Ubiquitous Comput.*, 6(3), 220-234.

- Smullen, C., & Smullen, S. (2006). *Modelling AJAX Application performance* Paper presented at the Web Technologies, Applications, and Services 2006, Calgary.
- Smullen, C., & Smullen, S. (2007). *AJAX application server performance*. Paper presented at the SoutheastCon, 2007. Proceedings. IEEE, Richmond, VA, USA, .
- Sonntag, M. (2006). *Ajax Security in Groupware*. Paper presented at the Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications.
- Stodel, E. J., Thompson, T. L., & MacDonald, C. J. (2006). Learners' Perspectives on What is Missing from Online Learning: Interpretations through the Community of Inquiry Framework. *The International Review of Research in Open and Distance Learning*, 7(3).
- Stuckmann, P., Ehlers, N., & Wouters, B. (2002). *GPRS traffic performance measurements*. Paper presented at the Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th.
- Theng, Y.-L., Tan, K.-L., Lim, E.-P., Zhang, J., Goh, D. H.-L., Chatterjea, K., et al. (2007). *Mobile G-Portal supporting collaborative sharing and learning in geography fieldwork: an empirical study*. Paper presented at the Proceedings of the 2007 conference on Digital libraries, Vancouver, BC, Canada.
- Traxler, J. (2005). *Defining Mobile Learning*. Paper presented at the Proceedings IADIS International Conference Mobile Learning 2005, Malta.
- Traxler, J. (2007). *Defining Discussing and Evaluating Mobile Learning : the moving finger writes and having writ...* Paper presented at the The International Review of Research in Open and Distance Learning.
- Vaishnavi, V., & Kuechler, B. (2007, 12 Oct 2006). Design Research in Information Systems. from <http://www.isworld.org/Researchdesign/drisISworld.htm>
- Vodafone New Zealand. (2007). Our Coverage. Retrieved 5th August, 2007, from <http://www.vodafone.co.nz/help/coverage/our-coverage.jsp>

- W3C.org. (2007a). Cascading Style Sheets. Retrieved 2nd August, 2007, from <http://www.w3.org/Style/CSS/>
- W3C.org. (2007b). Document Object Model (DOM). Retrieved 3rd August, 2007, from <http://www.w3.org/DOM/>
- W3C.org. (2007c). Extensible Markup Language (XML). Retrieved 3th August, 2007, from <http://www.w3.org/XML/>
- W3C.org. (2007d). XHTML2 Working Group Home Page. Retrieved 5th August, 2007, from <http://www.w3.org/MarkUp/>
- W3C.org. (2007e). The XMLHttpRequest Object. Retrieved 15th August, 2007, from <http://www.w3.org/TR/XMLHttpRequest/>
- W3Cschools.com. (2007). Introduction to XML. Retrieved 4th August, 2007, from http://www.w3schools.com/xml/xml_whatIs.asp
- WAP Forum. (2000, 20th Oct 2007). Wireless Markup Language Specification. from http://www.openmobilealliance.org/release_program/docs/Browsing/V2_1-20050614-C/WAP-191-WML-20000219-a.pdf
- WAP Forum. (2001, 20th Oct 2007). XHTML Mobile Profile. from <http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf>
- WAP Forum. (2002a). Wireless Application Protocol Architecture Specification. Retrieved 14th Oct, 2007, from <http://www1.wapforum.org/tech/terms.asp?doc=arch-30-apr-98.pdf>
- WAP Forum. (2002b, 16th August). Wireless Application Protocol WAP 2.0 Technical White Paper. from www.wapforum.org/what/WAPWhite_Paper1.pdf
- Watson, H., Rainer, R., & Koh, C. (1991). Executive Information Systems: A Framework for Development and a Survey of Current Practices. *MIS Quarterly*, 15(1), 13-30.

- Wei, C. K. (2005). AJAX: Asynchronous Java + XML? Retrieved March 4th, 2007, from <http://www.developer.com/design/article.php/3526681>
- White, A. (2005). Measuring the Benefits of Ajax. Retrieved 15th August, 2006, from www.developer.com/java/other/article.php/3554271
- Yokomoto, C. F. (2000). *Promoting depth of knowledge through a new quiz format to improve problem solving abilities*. Paper presented at the Frontiers in Education Conference, 2000. FIE 2000. 30th Annual.
- Zhang, Y., & Soong, B.-H. (2004). *Multi-slot capability increases GPRS network performance*. Paper presented at the Wireless Communication Systems, 2004. 1st International Symposium.

Appendix 1 — Bibliography

1. New Web-based Technology Draws Applications, Investors. Wall Street Journal: (2005-11-03).

http://online.wsj.com/public/article/SB113098635587487074-3diFzslPm_iutdYLU2C5e4DinUA_20061103.html

2. Foundations of Ajax

by Ryan Asleson, Nathaniel T. Schutta

3. An Introduction to Ajax

http://dev2dev.bea.com/pub/a/2005/08/ajax_introduction.html

4. Is Latency the Real Enemy in Next Generation Networks?

http://www.deutsche-telekom-laboratories.de/~vidales/pubs/talk_conwin2005_cottingham.pdf

5. Flow of Data through WAP and TCP/IP Protocol Stacks.

<http://www.it.iitb.ac.in/xnet/wap/default/protocol2.html>

6. WAP2.x architecture – Features, Services and functions.

http://www.ericsson.com/ericsson/corpinfo/publications/review/2001_04/files/2001043.pdf

7. GPRS (General Packet Radio Service), HSCSD & EDGE

<http://www.mobile-phones-uk.org.uk/gprs.htm>

8. What does Coding and Timeslots have to do with GPRS and 3G UTMS

http://www.gsmmobile.co.nz/GPRS_Time_Slots.htm

9. Google for Mobile (XHTML+XML Website)

<http://www.google.co.nz/m>

10. Vodafone New Zealand WAP Website

<http://live.vodafone.co.nz>

12. XHTML tutorial

<http://www.w3schools.com/xhtml/default.asp>

13 Before and After: How Can You Show Your New Website is Better?

<http://psychology.wichita.edu/surl/usabilitynews/2S/compare.htm>

14. Bluetooth mobile as a GPRS modem for laptop and desktop - fast configuration
<http://www.geekzone.co.nz/content.asp?contentid=1524>

15. Dynalink USB GPRS Mobile Modem Review
<http://www.geekzone.co.nz/content.asp?contentid=668>

16. How to connect your laptop to the Internet using your GPRS mobile via Bluetooth - full configuration
<http://www.geekzone.co.nz/content.asp?contentid=650>

Appendix 2 — Glossary of Term & Acronyms

Ajax — Asynchronous JavaScript and XML

ASP — Active Server Pages

CDMA — Code Division Multiple Access

CSS — Cascading Style Sheets

DTD — Document Type Definition

GPRS— GPRS (General Packet Radio Service) is the world's most ubiquitous wireless data service, available now with almost every GSM network.

GSM — GSM (Global System for Mobile communications) is the technology that underpins most of the world's mobile phone networks.

HTML — Hypertext Markup Language

HTTP — Hypertext Transfer Protocol

IEEE — Institute of Electrical and Electronics Engineers, Inc, a non-profit organization, IEEE is the world's leading professional association for the advancement of technology.

IIS — Internet Information Services

JavaScript — JavaScript is the scripting language of the Web.

JSON — JSON (JavaScript Object Notation) is a lightweight data-interchange format.

Microsoft Ajax 2.0 — ASP.NET AJAX is a free framework for quickly creating efficient and interactive Web applications that work across all popular browsers.

MMS — Multimedia Messaging Service

PDA — Personal Digital Assistant

PHP — PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

PSTN — Public Switched Telephone Network

SMS — Short Message Service

TCP/IP— Transmission Control Protocol / Internet Protocol, is the communication protocol for the Internet.

W3C — The World Wide Web Consortium (<http://www.w3c.org>) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential.

WAP 1.0 — Wireless Application Protocol version 1.0, protocol was designed to show Internet-contents on wireless clients, like mobile phones.

WAP 2.0 — Wireless Application Protocol version 2.0

WML — Wireless Markup Language, is the language used to create pages to be displayed in a WAP browser.

WWW —World Wide Web

XHTML — Extensible HyperText Markup Language is a combination of HTML and XML, aim to provide stricter and cleaner version of HTML. It is a W3C Recommendation.

XHTML MP — XHTML Mobile Profile

XML — Extensible Markup Language is a markup language much like HTML, allow you to define your own tags, to describe data.

Appendix 3 — Usability Testing (Invitation Letter)

Dear Colleague:

This is an invitation letter to participate in usability testing for the “Applying Ajax to A Mobile Learning system” project.

There are two Mobile Learning Websites on the Internet have the same layout and user interface. Please visit them via I-mate SP5 smart phone, think aloud your feeling and answer some questions. So, there are two tasks in this usability test.

Task one: Finish all the questions in Website one by I-mate SP5 smart phone.

Task two: Finish all the questions in Website two by I-mate SP5 smart phone.

Don't worry about the correct answer. We are not collecting or saving any answer from the Website. You can select any answer you want. And don't worry about the task finish time. You can take as long as you want to finish the task.

This is a low risk anonymous usability test. Your answers will be completely confidential. Thank you in advance for your participation. . If you have any questions about this test, please ask or contact me. xiefeng401@gmail.com

Yours Sincerely

Felix Xie

Institute of Information & Mathematical Sciences

Massey University at Albany, Auckland, New Zealand

Appendix 4 — Usability Testing (Checklist)

Assessing the Benefits of AJAX in Mobile Learning Systems

Design

Usability Testing Process Checklist (Anonymous)

1. Invite the evaluator (invitation letter).
2. Introduce the evaluator to goal of this usability test.
3. Introduce the evaluator the equipment need to be used (I-Mate SP5 Smart).
4. Advise the evaluator the Task he/she needs to finish.
5. Short tutorial about the task. (PowerPoint)
6. Short tutorial for the evaluator to get familiar for the phone. (Using I-mate SP5 to finish Website Demo questions.)

Evaluation Start

7. Background of evaluator (Section A questions).
8. Start task one.
(Answer any questions they ask)
9. Task one finished. (check time)
10. 5 mins break
11. Start task two.
(Answer any questions they ask)
12. Task two finished. (check time)
13. Post-task walkthroughs. (Section C: Questionnaire)
14. Thank you to evaluators for participating.
15. Prepare the mobile for next evaluator. (clean the cache)

Appendix 5 — Usability Testing (Questions)



Massey University

Te Kunenga ki Pūrehuroa

Applying Ajax to a Mobile Learning System

Think Aloud Usability Testing

(Official only)

SECTION A (Before the Task):

- 1. Gender: (Please cycle) Male Female.**
- 2. What kind of mobile phone do you have?**

- 3. Have you ever used your mobile phone to access the Internet?**

If so, how often?

SECTION B (During the Task):

- 4. Task Finish time**

Task one (Website 1): Start: _____ Finish: _____

Task two (Website 2): Start: _____ Finish: _____

SECTION C (All tasks finished):

Goal: After visiting two mobile learning Websites, we would like to hear your responses to using these Websites (post-task walkthrough).

- 6. Did you notice any differences between these two Websites?**

If yes, what is it?

- 7. Which Website you prefer to use? (Please cycle)**

	Website One	Website Two
Why?		

Appendix 6 — Participants Profile

ID	Gender	Mobile Phone	Access Internet	Task One Time	Task Two Time	Notices Different	Prefer
1	Male	Nokia N70	Once a Week	90	60	Yes	2
2	Male	Nokia 6210	No	120	65	Yes	2
3	Male	Panasonic v3	Just once	105	90	No	2
4	Male	No	No	60	56	Yes	2
5	Female	Sony Ericsson	No	115	62	Yes	2
6	Male	Motorola	Few times	126	---	---	---
7	Male	Motorola	No	150	66	Yes	2
8	Female	Nokia	No	156	77	No	0
9	Female	Sony Ericsson	few times	120	70	Yes	2
10	Male	Nokia	Just once	110	75	Yes	2
11	Male	O2	Once a Week	90	50	Yes	2
12	Male	Siemens S65	No	89	56	Yes	1
13	Male	Sony Ericsson	few times	110	43	Yes	2
14	Female	Sony Ericsson	No	116	60	No	0
15	Male	Samsung	No	133	50	Yes	2
16	Female	Sony Ericsson	Just once	130	75	No	1
17	Female	Nokia	No	120	60	Yes	0
18	Male	Nokia N73	few times	113	52	Yes	2
19	Male	Sony Ericsson	No	160	90	No	1
20	Male	Samsung	No	135	90	Yes	2

*which web site user prefer to use, “1” means Website one, “2” means Website2, “0” means no prefer Website.

ID	Reason**
1	Faster
2	Response time is different and the web two has quick response time.
3	Less response time
4	The second one seems to be fast, and the options appear more quickly than the question.
5	Second one fast than first one.
6	Didn't finish the test, because of his don't have enough time and the bad navigation button.
7	Faster
8	Almost the same, The Navigation button cause big problem in the test,
9	Faster
10	Bad Navigation button, fast response time.
11	Website is faster, bad button
12	Can see the website one loading, Bad button
13	Second one fast than first one.
14	Bad Navigation button, None prefer
15	Second one fast than first one.
16	notice is loading
17	Loading message position is different, one on top, one on the button.
18	Loading type is different, web one need to load every time press the link. Web 2 don't
19	Bad button, web one load faster.
20	Faster

** Which website they prefer and why, and what they had said during the test or after the test.

Appendix 7 — All Multi-Choice Questions

All Questions List

ID	question	explanation
1	$1 + 1 = ?$	$1+1=2$
2	$3 + 5 = ?$	$3+5=8$
3	$3 + 6 = ?$	$3+6=9$
4	$2 + 3 = ?$	$2+3=5$
5	$7 + 1 = ?$	$7+1=8$
6	$1 + 3 = ?$	$1+3=4$
7	$2 + 1 = ?$	$2+1=3$
8	$2 + 2 = ?$	$2+2=4$
9	$4 + 4 = ?$	$4+4=8$
10	$3 + 3 = ?$	$3+3=6$