View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by South East Academic Libraries System (SEALS)

A Framework for Responsive Content Adaptation in Electronic Display Networks

A thesis submitted to Rhodes University
in fulfilment of the requirements
for the degree of Master of Science

Philip West
Department of Computer Science
Rhodes University

2005

ABSTRACT

Recent trends show an increase in the availability and functionality of handheld devices, wireless network technology, and electronic display networks. We propose the novel integration of these technologies to provide wireless access to content delivered to large-screen display systems. Content adaptation is used as a method of reformatting web pages to display more appropriately on handheld devices, and to remove unwanted content. A framework is presented that facilitates content adaptation, implemented as an adaptation layer, which is extended to provide personalization of adaptation settings and response to network conditions. The framework is implemented as a proxy server for a wireless network, and handles HTML and XML documents. Once a document has been requested by a user, the HTML/XML is retrieved and parsed, creating a Document Object Model tree representation. It is then altered according to the user's personal settings or predefined settings, based on current network usage and the network resources available. Three adaptation techniques were implemented; spatial representation, which generates an image map of the document, text summarization, which creates a tree view representation of a document, and tag extraction, which replaces specific tags with links. Three proof-of-concept systems were developed in order to test the robustness of the framework. A system for use with digital slide shows, a digital signage system, and a generalized system for use with the internet were implemented. Testing was performed by accessing sample web pages through the content adaptation proxy server. Tag extraction works correctly for all HTML and XML document structures, whereas spatial representation and text summarization are limited to a controlled subset. Results indicate that the adaptive system has the ability to reduce average bandwidth usage, by decreasing the amount of data on the network, thereby allowing a greater number of users access to content. This suggests that responsive content adaptation has a positive influence on network performance metrics.

ACKNOWLEDGEMENTS

Thank you to my supervisors, Dr Greg Foster and Professor Peter Clayton, for their guidance, enthusiastic involvement, support, and general hard work, during the course of this research. Without their help this work would not have been possible.

I would also like to thank my parents for their support and encouragement during difficult times. The morale support offered by my colleagues and friends, Angus Black, Jason Penton, Jake Opie, and Adam King, is greatly appreciated.

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, and I would like to acknowledge the financial support from Telkom SA, Business Connexion, Verso Technologies, THRIP, the National Research Foundation, and Microsoft.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

Chapter 1

INTRODUCTION

Advances in wireless network technology and mobile device technology drive internet access via mobile devices. However, these technologies have unique considerations when compared to wire-line networks and desktop computers, and thus require novel approaches to handling content. Adaptation techniques attempt to solve this issue by manipulating content documents to display more appropriately on small-screen mobile devices. Digital signage is fast becoming the norm for displaying information and advertisements in public areas. The proliferation of digital signage, wireless and mobile device technologies drives the opportunity to expose digital signage content to mobile devices. Content adaptation techniques provide a possible solution for displaying the large digital signage displays on the comparatively smaller mobile device screens.

The work done during this study identifies an opportunity created by the technological advance and availability of electronic display systems, wireless networks and mobile devices. This Chapter provides a brief overview of these technologies, and describes how each technology contributes towards the motivation for this study. A discussion on electronic display systems creates the platform for the study, describing the growth and functionality of these approaches. With regard to wireless networking, emphasis is placed on the 802.11 group of standards, as these technologies are the most prevalent wireless local area network technologies in use in South Africa. However, other wireless technologies are also described as the work is not restricted to the 802.11 group of standards. The unique issues introduced by mobile devices are then examined, and serve as a basis for developing content adaptation solutions. Information gained from the three fields of electronic display systems, wireless networks, and mobile devices, is summarised as the motivation for the study. A thesis problem statement is then formulated, describing the aims of this project. Finally, the structure of the study is provided.

1.1. Supporting Technologies

1.1.1 Electronic Display Systems

Electronic display systems are designed to convey information to observers in a digital, multimedia environment, which is usually presented using a large screen device, such as a plasma screen television or a projector. A wide variety of implementations exist, including systems for corporations, public venues, retail outlets, and education (Omnivex Corporation 2005). Corporate implementations provide critical information for staff members, and can include presentation systems. Public venues include sports stadiums and airports, and these systems provide information and advertising to the public. Retail outlets are either single shops or entire shopping centres, and also provide information and advertising. Electronic display systems for educational environments usually consist of a presentation screen, and provide educational or instructional information to students. Information displayed using electronic display systems is generally not accessible via mobile device access, and the opportunity exists to expose this data to clients (Harrison & Andrusiewicz 2004).

Advertisers frequently employ conventional, static signage techniques as part of an advertising campaign. Advertisements are placed on large, roadside billboards, on bus stops, indoors in shopping malls, airports and hotels. These static signage approaches are not limited to advertising, and can be used for the distribution of information. The majority of the cost of conventional signage is the cost of materials used to construct the sign (creation), the transport required to deliver signs (distribution), and the labour used to set up the signs (installation). A further drawback is the slow end-to-end process of creation and installation, which may result in missed revenues due to missed opportunities (Harrison & Andrusiewicz 2003). The proliferation of the internet in the 1990s presented new advertising opportunities. Since then, marketing and advertising technologies have been developed to address these possibilities. Examples include banner advertisements on web sites, pop-up windows, e-mail marketing and cookies. Digital signage extends traditional static signage by displaying content on large screen displays. This approach is known as narrowcasting. A digital signage network (DSN), consists of numerous display terminals controlled by a central server. A wide variety of organisations, such as airports and shopping malls, are employing digital signage systems as a means of displaying information and generating additional revenue through advertising. Digital signage exhibits various advantages over the conventional signage approach. Static signage displays a single image, which can not be quickly changed or adjusted to suit time or context. Digital signage allows rich, multimedia content to be displayed. This content can easily be changed, to increase the possibility of exposing potential viewers to a greater number of displays. In addition to providing more compelling signage displays, the digital approach reduces the cost of creation, distribution and installation, and reduces the time from creation to installation (Harrison & Andrusiewicz 2004).

1.1.2 A Brief Review of Computer Networks

Recent improvements in wireless networking technologies provide the possibility to allow mobile device access to content. There are numerous types of wireless networks, each offering different functionality and issues, and this section contains a brief overview of relevant wireless technologies. This project utilises wireless networking to provide mobile device access to content displayed on electronic display systems, and aims to combat certain issues associated with wireless technologies. In computer networking, Quality of Service (QoS) refers to a level of throughput that is guaranteed by a network operator, and describes the overall experience a user or application will receive over a network (Nortel Networks 2003). For the purposes of this project, we define QoS to be a level of bandwidth usage in a network. That is, a high QoS will attempt to maximise the bandwidth available to each client on the network.

Initially, computer systems were highly centralised. Advances in technology greatly increased the power, and reduced the size, of these early machines. The centralised model was replaced by one where many autonomous, connected computers carry out the work load. The major benefit of computer networks is resource sharing, and the goal is to make all programs, equipment, and especially data, available to anyone on the network without regard to the physical location of the resource and the user. There are two kinds of transmission schemes, broadcast and point-to-point (Tanenbaum 2003). In broadcast approaches, a single communication channel is shared by all computers on the network. A data packet is sent to all these machines and the recipient is identified using an IP address. Multicasting is a broadcast scheme where packets are delivered to a subset of machines on the network. Point-to-point networks consist of many connections between individual pairs of computers. Packets traverse the network by following paths between computers. This approach is known as unicasting. Networks can be divided into local, metropolitan, and wide area networks depending on the scale of the network. The internet is a network of computer networks using common protocols (Tanenbaum 2003). Computer networks were readily adopted in early 1980s. The passing and sharing of data through a computer network increased a computer's usefulness and increased efficiency in research and productivity. These improvements drove the widespread adoption of networks. Varying network standards and technology made it difficult to maintain and administer the networks, as many vendor products were incompatible. Standards groups among leading manufacturers collaborated to ensure that hardware from different vendors could function together. The Transmission Control Protocol/Internet Protocol (TCP/IP) was developed to allow computers to connect across different cities and campuses. The Simple Network Management Protocol (SNMP) was widely adopted to manage network components.

Wireless networks (WLANs) were readily adopted in the late 1990s. WLANs had been in existence since the 1980s but mainly for specialised applications. Two main advances contributed to the mainstream adoption of wireless networks. The first was the advances in data throughput rates and radio size that increased the viability of using WLANs for general purpose applications. The second was the development of standards by the Institute of Electronic and Electrical Engineers (IEEE)[i]. This facilitated the interoperability of devices originating from different vendors (Harada 2005).

1.1.2.1 Wireless Standards

Standards play a vital role in wireless network technology by ensuring interoperability of products, providing economies of scale thereby reducing cost, and reduce investment risk for operators. The IEEE standards for wireless networks cover

[i] http://www.ieee.org/

specifications for all types of network configurations (Marks 2002). The 802.15 specification, known as Bluetooth, defines a wireless standard for Personal Area Networks (PANs), the 802.11 specifications, known as WiFi, define wireless standards for Local Area Networks (LANs), and the 802.16 specification defines a wireless standard for Metropolitan Area Networks (MANs). IEEE 802.20 is a proposed specification aimed at providing a wireless standard for Wide Area Networks (WANs). The European Telecommunications Standards Institute (ETSI)[ii] is the body responsible for creating telecommunications standards in Europe. The ETSI counterparts to the IEEE wireless standards are ETSI Hiperpan at the PAN level, ETSI Hiperlan at the LAN level, ETSI Hiperman at the MAN level, and ETSI is a partner of the Third Generation Partner Project (3GPP)[iii] and the Enhanced Data rates for GSM Evolution (EDGE) at the WAN level. Both the IEEE and ETSI standards are summarised in Figure 1-1. The IEEE standards are summarised below, and have been selected for use with this project owing to the prevalence of these technologies in South Africa.

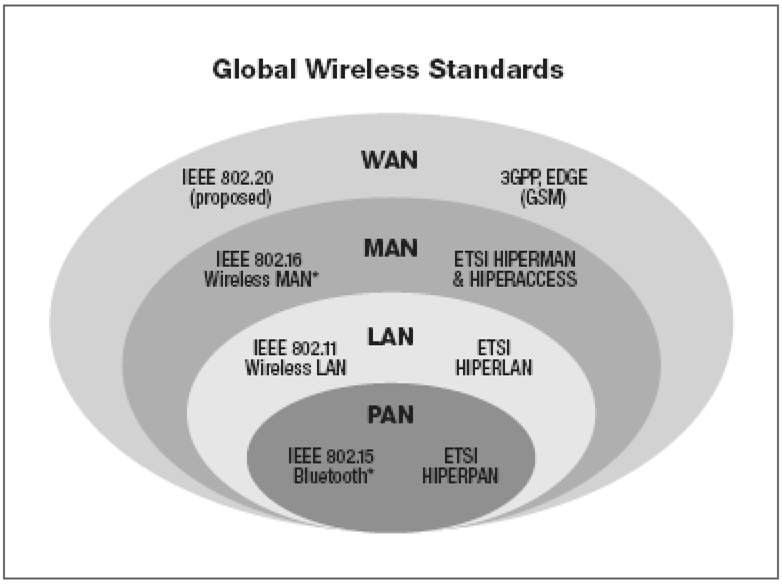


Figure 1-1. Global wireless standards showing the 802.x standards and the corresponding ETSI standards (Intel 2003).

[ii] http://www.etsi.org/
[iii] http://www.3gpp.org/

1.1.2.2 802.15 - Bluetooth

In 1994 engineers at Ericsson Mobile Communications investigated the possibility of using a low-cost, low-power radio interface to allow mobile telephones to communicate with mobile accessories. The Bluetooth Special Interest Group[iv] was formed in 1998 and released the first version of the specification in 1999. Bluetooth allows electronic devices to interconnect and communicate, eliminating the need for cables. The standard facilitates both data and voice communications, allows for ad hoc networks to be created, smart device recognition and data synchronisation. Infrared has been used in a similar way, but suffers from two major disadvantages. Firstly, infrared relies on line-of-sight. Any obstructions will interfere with the communication. Secondly, infrared devices can only communicate with a single device at one time. The Bluetooth standard works on a radio frequency (RF) and allows both half-duplex and full-duplex communication limited to 1 megabit per second (Mbps). Bluetooth allows devices to locate other devices and communicate with no user input. When this occurs the devices form a Personal Area Network (PAN), which is known as a piconet. Each piconet consists of a master device and up to 7 slaves. Two or more interconnected piconets form a scatternet. Bluetooth devices transmit signals at 1 milli-watt, which limits the range to 10 metres. This weak signal prevents the devices from interfering with other systems (Franklin 2005).

1.1.2.3 802.11 - WiFi

Wireless networks offer various advantages over wire-line networks. Users can be mobile, allowing them to access the network while on the move. It is difficult to install a wired network in certain locations, making wireless an easier, and possibly cheaper, option. Wireless networks are flexible, allowing users to easily be added, or the scope of the network to be increased. Small, temporary networks can be rapidly deployed. Initially, users of wireless networks were forced to adopt single vendor implementations. The completion of the 802.11 standard in 1997, coupled with advances in network technology, facilitated the widespread adoption of wireless networks.

[iv] http://www.bluetooth.com/

Wireless networks most commonly transmit data over either infrared or radio waves. Infrared is limited as a transport medium in that it is easily blocked by obstructions. Radio waves are capable of penetrating many common obstructions, and provide a wider coverage range than infrared. For these reasons most 802.11 wireless technologies implement radio waves in the physical layer. Wireless devices operate in a certain frequency band, with an associated bandwidth which is the amount of frequency space in the band and corresponds to the data capacity of the link. Radio waves pose several problems, including several propagation problems that may interrupt the radio link. These include multi-path interference and shadows. Regulatory associations allocate frequency bands in order to prevent overlapping uses of radio waves. Other band uses include television and radar networks. 802.11 devices are placed in a band category known as the industrial, scientific and medical (ISM) bands. The ISM bands are license free, as long as the devices are low-powered, meaning that a license is not required to deploy a low-powered wireless network. The speed of wireless networks is limited by the available bandwidth. Wireless network hardware is generally slower than wired network hardware. Wireless networks suffer from various unique security concerns. Unlike wired networks, where data is kept within the cables, wireless transmissions are available to any device within the broadcast range. Wireless networks tend to suffer from fuzzy boundaries, where a network broadcast range may extend outside the desired area (Gast 2002).

An 802.11 wireless network is comprised of four main components, a distribution system, a wireless access point, a wireless medium and a station (Figure 1-2). The distribution system is used to forward data to the correct location when numerous access points are used to provide coverage for a network. This system is typically implemented using Ethernet to create the backbone network technology. Access points provide bridging from wireless to wired network frames and vice-versa. A wireless medium is used to transmit frames between stations. The medium can be either one of two RF physical layers, or an infrared physical layer. The RF layers have been most widely adopted. Stations are computers with wireless network interfaces. The three most prominent versions of the 802.11 standard are 802.11b, 802.11a, and 802.11g. 802.11b was the first widely accepted wireless network standard, followed by 802.11a and g. 802.11b and g operate in the unlicensed 2.4GHz range, whereas 802.11a operates in the 5GHz range (Table 1-1).

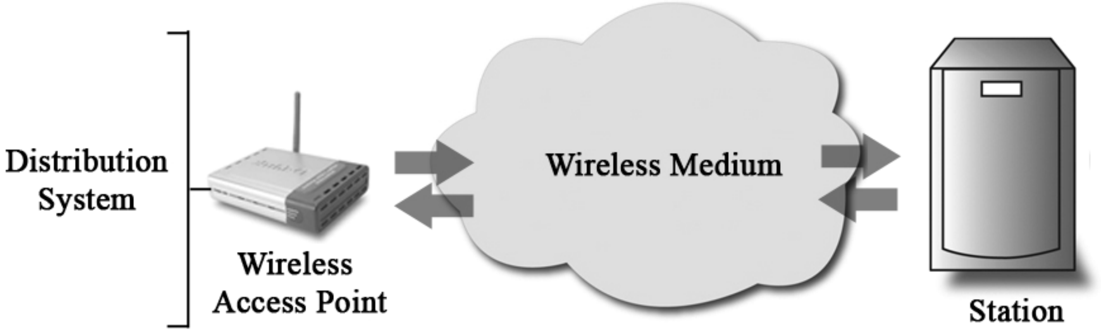


Figure 1-2. Components of an 802.11 wireless network.

Table 1-1. Comparison of 802.11 standards.

IEEE Standard	Speed	Frequency Band
802.11	1 Mbps 2 Mbps	2.4 GHz
802.11a	up to 54 Mbps	5 GHz
802.11b	5.5 Mbps 11 Mbps	2.4 GHz
802.11g	up to 54 Mbps	2.4 GHz

1.1.2.4 802.16 - WiMAX

The Worldwide Interoperability for Microwave Access Forum (WiMax) is a non-profit corporation formed by equipment and component suppliers, including Intel Corporation, to promote the adoption of IEEE 802.16 compliant equipment by operators of broadband wireless access systems. The organization is working to facilitate the deployment of broadband wireless networks based on the IEEE 802.16 standard by helping to ensure the compatibility and interoperability of broadband wireless access equipment. In this regard, the philosophy of WiMAX for the wireless MAN is comparable to that of the Wi-Fi Alliance in promoting the IEEE 802.11 standard for wireless LANs. The IEEE 802.16 standard is aimed at providing broadband wireless access using a metropolitan area network (MAN). This specification offers the same advantages as the 802.11 standards, and extends these to include greater range and transmission rates comparable with traditional cable, DSL, and T1 approaches. The typical 802.16 configuration consists of a mounted base-station communicating on a point to multi-point basis with subscriber stations. Broadcast ranges can reach up to 48 kilometres, with a typical cell radius of 6 to 10 kilometres. Within the cell radius, the standard provides non-line-of-sight performance and shared data rates of up to 75 Mbps. A single station can provide simultaneous T1 connectivity to up to 60 stations. 802.16 can provide services requiring low-latency, including voice and video. Voice services can be traditional Time Division Multiplexed (TDM) or Voice over IP (VoIP). The standard can be used to accelerate the deployment of 802.11 LANs in areas where cable or DSL is unavailable. The IEEE 802.20 proposed standard aims to provide wireless WAN access of up to 1 Mbps for vehicles travelling up to 250 Km/h (Lewis & Davis 2004).

1.1.2.5 Wireless Hotspots

Wireless hotspots provide network access to mobile device users, offering location specific services such as internet access (Minoli 2003). Hotspots can contain captive portals, where users are redirected to for authentication and payment purposes. The use of hotspots is becoming more prevalent in businesses. There is growing deployment in coffee shops, airports, restaurants and corporate environments. The benefit to businesses is two-fold: hotspots can increase productivity by providing wireless network access to employees, allowing access to e-mail and network resources; hotspots can also be installed as an extra service, creating product differentiation and attracting customers. According to (Molta 2003), there will be 31 million frequent and 35 million infrequent hotspot users by 2007 accessing 120,000 WLAN hotspots. The revenue from this hotspot usage is projected at 9 billion US Dollars.

1.1.2.6 Cellular Communication

An alternative form of wireless technology is cellular networks. Initially these systems provided only voice services, but recently many data services have become readily available. Coupled with the rapid increase in cellular telephone technology, cellular networks are a viable alternative to wireless radio networks, such as the IEEE and ETSI wireless standards. Wireless hotspots and 3G technologies can be viewed as competing technologies. However, the technologies are likely to coexist. Another viewpoint is that 3G will take the place of wireless hotspots, once data rates and coverage increase significantly. Hotspots are however likely to persist owing to the relatively low deployment operational cost involved. Cellular telephone vendors, such as Nokia, consider integration to be the next phase, with mobile devices supporting a range of the wireless technologies (Molta 2003).

Since their inception cellular systems have continued to attract an increasing number of subscribers. Bell Labs in the United States of America formed the concept of cellular communication. This was however first implemented in 1981 in the Nordic countries with the Nordic Mobile Telephone (NMT). In 1983 the Advanced Mobile Phone Service (AMPS) was developed in America, and was adopted by countries in Asia and Latin America. These early systems were analogue implementations and suffered from setbacks including an inability to handle growing capacity requirements in an economical manner. Digital systems offer numerous advantages over their analogue counterparts: ease of signalling, lower levels of interference, and increased ability to meet capacity demands. The 1980's witnessed rapid growth in the deployment of first generation (1G) analogue cellular telephone systems. However, a lack of standardisation lead to the creation of the Groupe Special Mobile (GSM) in 1982, and the system is now known as the Global System for Mobile Communications (GSM). In 1989 GSM was transferred to ETSI, and the first part of the GSM specifications were published in 1990. GSM implemented a digital system over the standard analogue systems in place at the time, making GSM a second generation (2G) technology. A GSM network is divided into three main segments: the switching system, responsible for handling call-processing and subscriber related functions, the base station system, which handles all radio-related functions, and the operation and support system, the network entity from which the network operator monitors and controls the GSM system. A cell is the area given coverage by a single base transceiver station. The most basic service supported by GSM is telephony, known as teleservices. Here speech is digitally encoded and transmitted as a digital stream. GSM supports a wide variety of data services, also known as bearer services. These include the short message service (SMS) and fax services. GSM also supports a range of supplementary services that support and complement both teleservices and bearer services. These include call forwarding and call waiting (International Engineering Consortium 2005). The widespread adoption of GSM as a standard promotes international roaming. In 1998 the Third Generation Partnership Project (3GPP) was formed to create specifications for third generation (3G) networks. 3GPP also took over the responsibility of developing the GSM standard, with ETSI remaining as a 3GPP partner. GSM technology currently has more than 1 billion subscribers worldwide, controlling about 70% of the market (Wikipedia 2005a).

1.1.2.7 Cellular Data Services

Mobile operators have been offering data services for many years, including providing internet access. The two main approaches have been High Speed Circuit Switched Data (HSCSD) and the wireless application protocol. HSCSD provides many drawbacks owing to its circuit switched nature, once a connection is established it is necessary to maintain the connection for the entire lifetime and reserve the full bandwidth of the circuit. With this approach it is not realistic to provide a continuous connection because of the costs involved in establishing and maintaining a connection in a circuit switched network. The introduction of the General Packet Radio Service (GPRS) allows data packets to be transmitted on a GSM network on a packet-switched basis. GPRS is a cost-effective way to provide data services to mobile devices, providing an always connected architecture where fees are charged based on the amount of data transmitted, not the length of the call as is the case with circuit switched approaches. Data transmission rates can reach a theoretical maximum of 171.2kbps but, because GPRS is a packet-data overlay to GSM and must therefore compete for limited available radio frequency resources, actual achieved rates are much lower (50kbps) (Lin *et al.* 2002). GPRS is a 2.5 generation technology, providing moderate data transfer rates. GPRS data is usually billed per kilobyte of data, whereas circuit switched data is billed per second. Since GPRS data must share bandwidth, data rates can be low in busy cells. Enhanced Data rates for GSM Evolution (EDGE), a 2.75G technology, provides an improvement to 2.5G GPRS networks, and is also known as EGPRS. EDGE provides increased data transfer capabilities to a GPRS network, offering up to 384 kbps.

Third generation (3G) technologies provide both voice and data services at high transfer rates, making applications such as video telephony possible, and should provide seamless global roaming. The International Mobile Telecommunications-

2000 (IMT-2000) specification provides the global 3G technology standards as described by the International Telecommunications Union (ITU)[v]. Various 3G implementations exist, such as the Universal Mobile Telephone System (UMTS), which is based on Wideband Code Division Multiple Access (W-CDMA) technology and is the preferred 3G standard for GSM networks (Wikipedia 2005b).

1.1.3 A Brief Review of Mobile Devices

Rapid advances in mobile device technology have resulted in an increased adoption of mobile devices. Many of these devices are equipped with a form of wireless networking technology, such as Bluetooth and/or Wi-Fi, allowing access to network resources (Ridgeway 2002). However, the majority of content documents are designed for viewing on desktop computers or large screen devices, and not on the relatively smaller screens of mobile devices. Thus, in order to transform content to be more suitable for mobile devices, it is necessary to evaluate the unique requirements of these devices.

Mobile devices can be broadly categorised into mobile phones, personal digital assistants (PDAs), or laptop computers. Each group contains a wide variety of devices with a range of capabilities. Devices are placed into categories by examining screen size, connectivity, memory and processing power. Mobile phones are the most common mobile devices, providing primarily voice capabilities. Secondary functions include sending text messages, e-mail, and limited web browsing using the Wireless Application Protocol (WAP). Mobile devices are not ideally suited for the secondary functions owing to the small screen size and text entry systems. PDAs were originally developed as digital calendars, allowing schedule planning and contact information to be stored. Recently, the increase in processing power and addition of wireless capabilities has broadened the functionality of these devices. Increased processing power allows more complex applications to be developed for these devices, and additional benefits include features such as handwriting recognition. Wireless connectivity extends the capabilities of these devices to include e-mail applications and web browsing. Laptops provide a much larger screen size and processing power than either of the other device categories. The addition of a wireless network interface

[v] http://www.itu.int/home/

card provides wireless connectivity. Advances in technology have resulted in the merger of certain devices. For example, a PDA is now available that provides voice capabilities (Milroy *et al.* 2002). Mobile computers account for 25% of the computers in use today, almost doubling from 13.7% in 1995. This number is projected to increase further to 36% by 2010 (Computer Industry Almanac 2005).

Web technologies have become widely available, and a variety of content and services is readily accessible from desktop computers. Mobile devices have the potential to provide the same level of access to these web services. However, mobile devices suffer from certain issues that restrict this from occurring (W3C 2005b). These issues include:

- Bandwidth – Wireless access often offers lower bandwidth than fixed-line implementations.
- Battery – Battery life is limited, mainly owing to the small size of mobile devices.
- Capabilities – The large variety of mobile devices has resulted in a large range of device capabilities. These different capabilities make the creation of a standard web browser a difficult task.
- Input – Mobile device input methods are usually more inhibited than desktop computers. Input methods include predictive text, stylus, and touch screens. Mobile device character sets may also be more limited than with the desktop computers.
- Memory – Mobile devices generally offer less memory and storage space than desktop computers, again, owing to the small size of the devices.
- Processing power – The processing power is limited in the same way as the memory supplied. The lack of power requires software efficiency.
- Screen size – The smaller screen size makes viewing documents designed for desktop computers an issue. There is also a large variation in the screen sizes, resolutions and colour densities of these devices, introducing further presentation and navigation complexities.

The Mobile Web Best Practices Working Group (BPWG), a body founded by the World Wide Web Consortium (W3C)[vi], is tasked with developing "a set of technical best practices and associated materials in support of development of web sites that provide an appropriate user experience on mobile devices" (W3C 2005a). BPWG propose an "adaptation layer", which will be used to convert content to according to device capabilities. The adaptation layer is a transparent part of the web, functioning without end-user knowledge or input. Content should be created in a way that facilitates the functionality of the adaptation layer, and makes the results of the layer predictable. An approach such as this could have positive influences on the bandwidth, capabilities, memory, processing power, and screen size issues listed above.

1.2. Motivation for the Study

Electronic display systems are fast becoming the norm for displaying information and advertisements in public areas. Wireless local area networks (WLANs) are increasingly being deployed to support traditional wired LANs. Current implementations of the 802.11g standard deliver up to 54Mbps, which is comparable to the performance of standard wired local area networks (LANs). Current wireless technologies have associated problems, such as shared frequencies with other devices. However, technological advances will look to solve these issues as well as decrease the cost of installing WLANs (Internet Security Systems 2001). Recent computing trends have shown a world-wide increase in the use of mobile devices, and many of these devices are equipped with a form of wireless networking technology, such as Bluetooth and Wi-Fi. This, however, has not increased device size or reduced usability (Ridgeway 2002). Advances in wireless network technology and mobile device technology drive the possibility of developing new tools. However, these technologies have unique considerations when compared to wire-line networks and desktop computers, and thus require novel approaches to handling content. Different devices, with differing screen sizes and resolution, provide challenges for content providers and thus content should be easily adaptable. The proliferation of digital signage, wireless, and mobile device technologies drives the opportunity to expose

[vi] http://www.w3.org/

digital signage content to mobile devices. Content adaptation techniques provide a possible solution for displaying the large digital signage displays on the comparatively smaller mobile device screens. Content adaptation techniques have focused on representing data on small screen devices in a convenient way. These approaches have centred on manipulating web content. Content adaptation becomes a relevant tool owing to the small screen size of handheld devices, and the fact that content documents have been designed for viewing on large screens. W3C (2005a) proposes, and defines the need for, an adaptation layer that converts content based on device capabilities.

There has been an increase in the installation of DSNs, such as presentations, advertisements and notice boards. This, coupled with the increased adoption of mobile devices and the proliferation of wireless networks, provides an opportunity to provide unprecedented mobile access to content. Previous work relating to delivering digital signage content to mobile devices has only transmitted predetermined data, and not allowed the user to select the required content (Harrison & Andrusiewicz 2004).

1.3. Problem Statement

This project aims to develop a framework for exposing content, displayed on electronic display systems, to mobile device users. Various content adaptation techniques, implemented as an adaptation layer, will be employed in order to achieve this. These techniques will be evaluated according to effectiveness and performance. Effectiveness is evaluated as the correctness of the output produced by the adaptation approaches in terms of accuracy of representation, and is not a measure of usability. Further, we will extend the framework to include a system for personalising adaptation preferences, and provide a responsive adaptation approach, which will attempt to adjust adaptation settings according to network usage parameters. The adaptive system will be evaluated on its ability to improve the QoS in a wireless network. It should do this by reducing congestion on the network.

The aims of the framework, described in detail in Chapter 3, Section 3.2, are to provide:

- Wireless access to content for mobile devices.

- Content adaptation for appropriately displaying content on small-screen devices.
- An easy to use personalisation system for specifying user preferences.
- A network management approach allowing response to network conditions.
- Low response time overhead for all modules and extensions.
- A modular and extensible framework.
- Limited complexity for clients and servers.
- A generalised, flexible framework.

1.4. Structure of the Study

The first chapter provides an introduction to wireless networks, mobile devices, and electronic display systems. This introduction establishes the context of the work, and is made relevant by forming the basis of the motivation for the study. A thesis statement specifies the objectives of the work, and an overview of the study is provided that details the contents of each chapter.

Chapter two contains summaries of work that has been carried out in related fields, and any relevance to this project is discussed. The related work examined includes: electronic display system implementations, approaches used to adapt content to handheld devices, content personalisation approaches, and various methods for controlling network QoS at the data level. The outcomes of this chapter are used to establish the requirements for the adaptive framework.

Chapter three introduces the concept of extending electronic display systems to provide wireless access, and how this access is further facilitated by implementing content adaptation techniques. The requirements identified in the previous chapter are formalised. A framework is then presented that provides for these requirements. A broad overview of the approach is given, specifying the main goals. This framework provides the basis for the rest of the study, as all implemented systems adhere to the model.

Chapter four details the content adaptation techniques implemented during this study, examining the algorithms and implementation issues of each approach. The development of a system for personalising adaptation settings and preferences is described, and the positive and negative influences of this user-configurable approach are discussed. A responsive component to the content adaptation system that is capable of adjusting to network conditions is introduced. A method for monitoring network conditions is described, and adaptive logic is discussed.

Results obtained through testing the framework modules and extensions are presented and analysed in Chapter five, and the benefits and limitations of the implementations are discussed. Results include an evaluation of the content adaptation techniques implemented, the personalization approach, and the responsive adaptation system. These results are used to critically analyse the success of the framework modules and extensions in terms of effectiveness and performance.

Chapter six presents proof-of-concept implementations of the content adaptation framework detailed in chapter three, and the goal is to evaluate the proposed framework. The first implementation details a content adaptation system for digital slide show presentations. The second section describes the implementation of a digital signage system, based on XML. This system is then extended using the content adaptation and responsive adaptation approaches described in the framework. The third system is designed for web browsing, and is aimed to evaluate the framework for use with a wide variety of content documents.

The final chapter provides a summary of all findings and conclusions, and possible extensions to the work are suggested.

1.5. Summary

Electronic display systems, wireless networks, and mobile devices are becoming increasingly available and technologically advanced. Electronic display networks are used to distribute content to audiences in a wide variety of environments, from the corporate sector to public venues such as shopping centres and sports stadiums. This content is currently not accessible via mobile devices. The widespread adoption of wireless networking technologies and mobile devices provides the opportunity to extend electronic display networks to allow wireless, mobile device access. Wireless networks offer unique challenges compared to wire line networks, such as limited bandwidth. Content documents are generally designed to be viewed on desktop computers, or even large screen displays. The relatively small-screen size of mobile devices provides an issue for delivery of this content. Adaptation techniques attempt to address this issue by reformatting content to display more appropriately on mobile devices. These techniques are usually designed for transforming web pages. This project aims to develop a framework for exposing electronic display system content to mobile devices through a wireless network, and to address the issues inherent with mobile devices and wireless networks. Content adaptation modules will be developed in order to reformat content for display of small-screen devices. A personalisation system will be developed which allows mobile device users to specify adaptation settings. A responsive adaptation system will be developed which will monitor network conditions, and subsequently alter adaptation settings in order to maintain desired levels of network usage. Chapter two builds on this by discussing related work in the areas of electronic display systems, content adaptation techniques and systems, personalisation approaches, and systems for controlling network usage by manipulating content. Information obtained from these works is used as the basis for developing the adaptation framework.

Chapter 2

RELATED WORK

This project focuses on opportunities created by the proliferation of technological advances, and general availability of hardware, in the fields of electronic display systems, mobile devices, and wireless networks. Work in related fields of study is examined so as to ascertain the requirements of a framework for extending electronic display systems to allow mobile device access. Current electronic display approaches are examined in order to establish a generalised model, which will be used as a basis for designing an adaptation framework. The use of mobile devices to access content documents raises usability issues owing to the small-screen size of these devices. Methods of transforming or adapting documents for display on mobile devices are investigated as potential components in the framework. The architectures of these content adaptation systems are evaluated and form the basis for developing a responsive adaptation framework. Alternative content adaptation methods include personalising content documents based on user preferences, and are examined as a potential extension to the adaptation process. Content adaptation approaches can reduce document size, in bytes, and can therefore impact network usage. The feasibility of a content adaptation network management system is investigated by examining work done in this field.

2.1. Electronic Display Systems

Electronic display systems are being increasingly deployed worldwide. There are many current commercial digital signage solutions[vii], servicing a wide variety of environments including corporate, public venues, retail, and operations and call centres (Omnivex Corporation 2005). Many other forms of electronic display systems exist, including digital slide show presentation software, such as Microsoft PowerPoint[viii] and Open Office Impress[ix], and large-screen displays in sports

[vii] Examples of digital signage companies include: http://www.webpavement.com/; http://www2.smarttech.com/st/; http://www.digitalsignage-usa.com/; http://www.scala.com/; http://www.omnivex.com/products/digital_signage.asp

[viii] http://www.microsoft.com/office/powerpoint/prodinfo/default.mspx

stadiums. Electronic display networks typically consist of a central server, used to control remote electronic displays via a network (Figure 2-1). The network can be either wired or wireless. Content documents are designed either on the central server, or remotely and then transferred to the server. The display terminals receive the content, and instructions on when to display the content, from the central server. This is a generalised overview of the process and many variations of this approach exist.

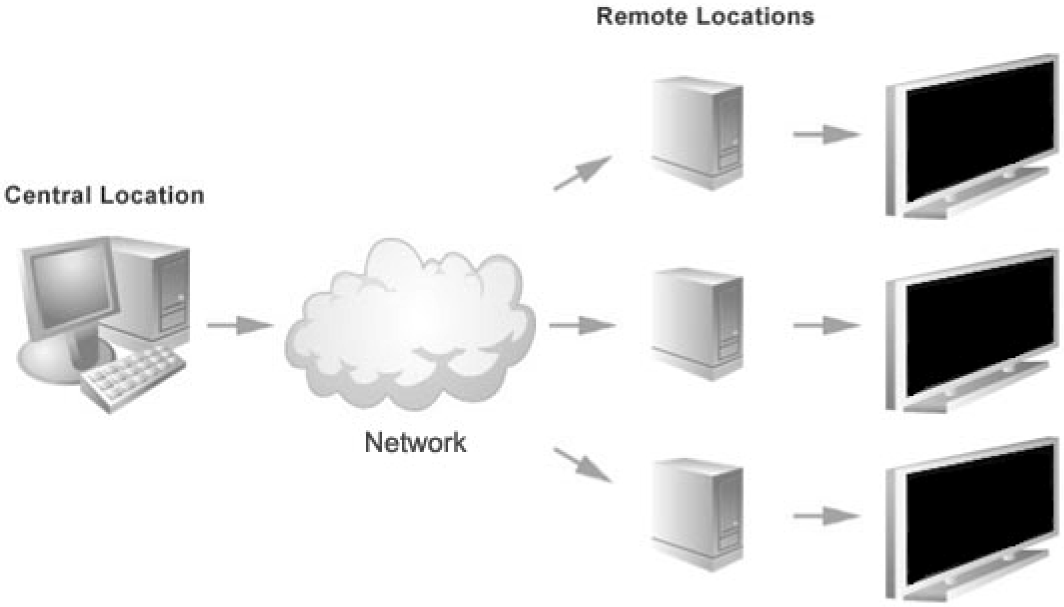


Figure 2-1. Electronic display network (Omnivex Corporation 2005).

2.1.1 Digital Signage Exchange

Harrison and Andrusiewicz (2003) developed a system known as the digital signage exchange (DSE). This approach facilitates the buying and selling of advertising space on a digital signage network (DSN) between the owner of the network and customers who wish to advertise using the network. The DSE is essentially a virtual marketplace that permits potential buyers to view and purchase all available advertising space on the DSN. The exchange provides potential buyers with relevant information regarding the network, including: the physical location of the display terminals; the type of venue, such as a movie theatre or shopping mall; demographics of the signs audience, such as age and income level; cost of the advertising space; possible commercial conflicts caused by a competitor advertising on the same terminal; advertising

[ix] http://www.openoffice.org/product/impress.html

restrictions of the venue, such as no advertising of alcoholic products. Advertising space can be sold using action techniques, brokering strategies and pricing strategies.

2.1.2 Enhancing Narrowcasting

Harrison and Andrusiewicz (2004) extend the traditional signage architecture by adding wireless access points to display terminals (Figure 2-2). A software application residing on the control computer handles connections between the client and the display terminal. The application monitors which information segment is displayed at the time of connection, and relevant documents are transferred to the requesting device. These documents may include a list of local vendors, digital coupons, or product information. A URL can be sent to the mobile device in instances where the document is too large to transmit. The client can then obtain the information when more bandwidth is available, or at a more convenient time. This approach provides functionality for users to order products directly, or give permission for vendors to contact them at a later stage. The system supports both 802.11b and Bluetooth connectivity, the type of protocol implemented depends on the density of display terminals and the expected amount of client traffic.

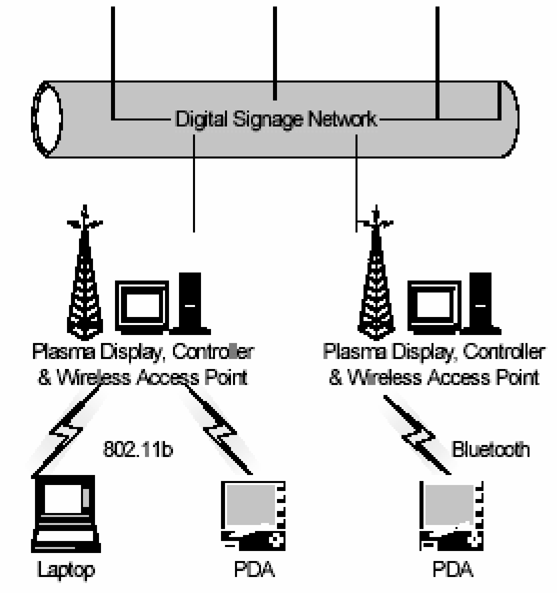
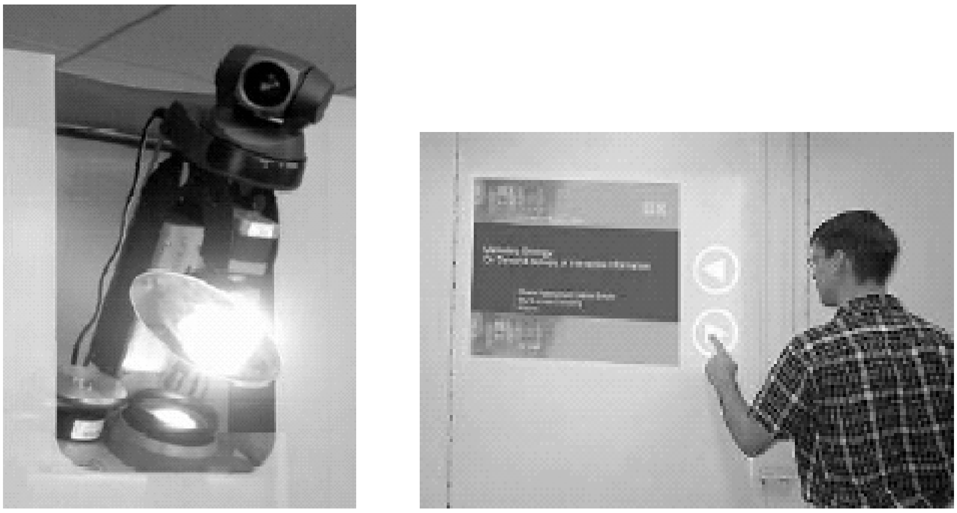

Digital Signage Network
Plasma Display, Controller & Wireless Access Point
Plasma Display, Controller & Wireless Access Point
802.11b
Bluetooth
Laptop
PDA
PDA

Figure 2-2. Augmented Digital Signage Network (Harrison & Andrusiewicz 2004).

2.1.3 Interactive Projected Displays

Pinhanez *et al.* (2003) present a touch-screen system for virtual-reality and digital signage approaches. The prototype system, called the Everywhere Display (ED), is comprised of a liquid crystal display (LCD) projector, and pan/tilt mirror and a camera (Figure 2-3a). The mirror is used to deflect the projection onto the required surface, such as a wall or the floor. The camera, along with a software application, is used to detect motion on the projected surface. Depending on the angle of projection, the captured images showing motion may need to be processed to compensate for perspective distortion. This approach removes the need for monitors and sensing devices used with conventional touch-screen applications. In contrast to traditional virtual reality systems, the need for goggles and specialised input devices is removed. The lack of a fixed monitor means that the display can easily be moved to a different surface by rotating the projector, mirror and camera. An example application of the system is to display a slide show presentation, allowing the user to scroll through the slides by touching associated images on the display (Figure 2-3b).



(a) **(b)**

Figure 2-3. (a) The Everywhere Display (ED) projector, camera and tilt/pan mirror. (b) An example of an interactive projected display (Pinhanez *et al.* 2003).

2.1.4 Discussion

Electronic display systems, at the most basic level, consist of a display terminal. An electronic display network is formed by linking multiple terminals via a network, and controlling these terminals from a central server. Electronic display networks are being readily deployed in many corporate and public environments. Display terminals consist of a computer, used to control the display, and a large screen. Currently, display terminals either do not allow client access to the content displayed, or only provide limited access, where pre-defined documents are transferred to the requesting device. Thus the opportunity exists to extend the display terminal implementations to provide mobile device access to content.

2.2. Content Adaptation

Multimedia content is becoming increasingly rich and diverse. At the same time, a wide variety of mobile devices with a wide range of capabilities exist. Rich multimedia content is mostly designed for viewing on desktop computers and large screen displays. Certain mobile devices suffer from restrictions when attempting to access these rich content pages (Chapter 1, Section 1.1.3). Various approaches have been developed in an attempt to solve some of these issues. Content adaptation refers to the retrieval of pertinent information from web pages. The aim is to remove aspects of the content document, in order to reduce size and complexity, while retaining the original meaning of the content. Content adaptation has positive implications for small screen devices, such as PDAs, and for network quality of service owing to the reduction in content size. Abdelzaher and Bhatti (1999) state that "web content must be adapted in a way that preserves essential information, yet reduces the resource requirements of content delivery". Content adaptation can take four forms: device-specific authoring, multiple-device authoring, client-side navigation, and automatic re-authoring (Bickmore & Schilit 1997). Device-specific authoring involves manually creating different content documents for devices of different capabilities. This approach creates accurate results, but is a time consuming process and may limit mobile device users to a subset of documents. Multiple-device authoring takes the same approach, but a content document covers a range of similar devices. However, this approach is still not ideal as a large amount of manual design is still required. Client-side navigation allows a mobile device user to interactively manipulate sections of a content document. Many of these approaches are limited in functionality and effectiveness. Automatic re-authoring takes a content document, designed for a desktop computer or other large screen display device, along with the mobile device capabilities and transforms the document to display conveniently on the device. Thus automatic re-authoring is the best approach for content adaptation as it provides the ability to effectively transform a broad range of content documents without the need for manually authoring alternative documents. The majority of content adaptation approaches focus on web content, and therefore mark-up languages.

HTML is currently the most prevalent language used to describe web content, and has long been the focus of content adaptation techniques. Mark-up languages are convenient for this purpose owing to the intrinsic structure used to define the semantics of a document. HTML can be converted to XHTML, which requires stricter structure. XML is currently the industry standard for specifying the information exchanged between applications across networks. XML allows software to interpret and manipulate data more effectively. Despite being a simple concept, the use of XML may reduce development costs, improve interoperability of data and increase the flexibility of applications (Feng & Zhu 2001). XML was designed to provide a structured description of documents and this functionality was extended to allow for the description of data structures sent across distributed environments. The communication capabilities of XML are more flexible and neutral than traditional techniques such as the Remote Procedure Calling Protocol (RPC) and the Common Object Request Broker Architecture (CORBA). Specifications are continuously being developed for improving security associated with XML.

2.2.1 Content Classification for Digital Libraries

Finn *et al.* (2001) propose a content extraction and classification scheme to address the issue of information overload in the web. A vast quantity of information is available on the internet, but retrieving and classifying documents poses unique problems. Information on the web is not always persistent. News articles, for example, may be available for short and undetermined lengths of time. The ability to locate and retrieve relevant information for digital library users is a complex, but valuable service. A web service known as HYPPIA monitors news websites for articles. Once new articles have been detected, text classifiers are used to classify the articles. Links to these articles are stored in a central information repository for use by the digital library. Articles are classified by first extracting the main body of the text from the web page. The extraction technique is based on the assumption that web pages consist of one main article surrounded by irrelevant content. Documents are considered to consist of two tokens, a tag token and a word token consisting of plain text. The document is represented as a sequence of bits, and can then be viewed as the distribution of cumulative tags as a function of position (Figure 2-4). The main body of text is considered to be the section of the document where the cumulative tag value stays constant over the greatest number of positions. This is illustrated in Figure 2-4 as the area between point *a* and point *b*.

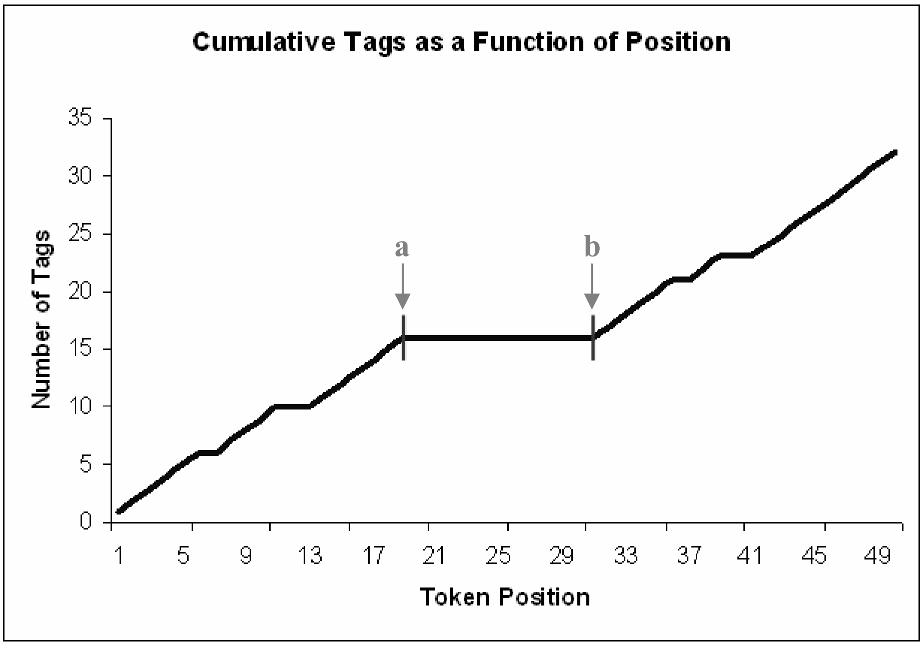


Figure 2-4. A visual representation of cumulative tokens as a function of position in a document.

2.2.2 Text Summarization

Buyukkokten *et al.* (2001) introduce a technique referred to as text summarization which improves access times of handheld device users and reduces the number of required pen actions while browsing the internet. The approach displays web pages only in text, and allows sections to be hidden, partially displayed, summarised or displayed in full. This system acts as a web proxy, which filters irrelevant content and transforms web pages into a suitable format for PDAs. A web page is summarised on a macro and micro-level. First, a page is summarised at the macro-level by structural analysis. Page partitions, known as Semantic Textual Units (STUs), are identified by paragraphs, lists, or tags that describe images. STUs are then arranged into a hierarchy using font information and structural information such as nested tags. Micro-level summarisation then identifies an efficient and convenient method for displaying each STU in the hierarchy. Five methods of micro-level summarisation have been developed. Incremental summarisation reveals an STU in three stages: the first line, then the first three lines and finally the entire STU. All summarisation refers to no summarisation, where the entire STU is displayed in a single state. Keyword summarisation displays keywords in the first state that have been identified in the STU. The next state shows the first three lines and after this the entire STU is displayed. The first state of the summary approach displays the most significant sentence in the STU, and the second state displays the entire section. A hybrid technique of keyword and summary initially displays only the keywords. The second state displays the most significant sentence, and finally the entire STU is shown.

2.2.3 Summarization of Graphical Displays

Chen *et al.* (2003) extend the text summarization method to include a solution for handling graphics. The web page is arranged into a two-tier hierarchy. The first tier provides an overview of the web page in the form of a graphical table of contents, which is displayed as a thumbnail. Content is grouped according to semantic relation, and these groupings are then displayed on the thumbnail in different colours. In order to view content the user selects a colour block, which has been formatted for display on a small screen (Figure 2-5).

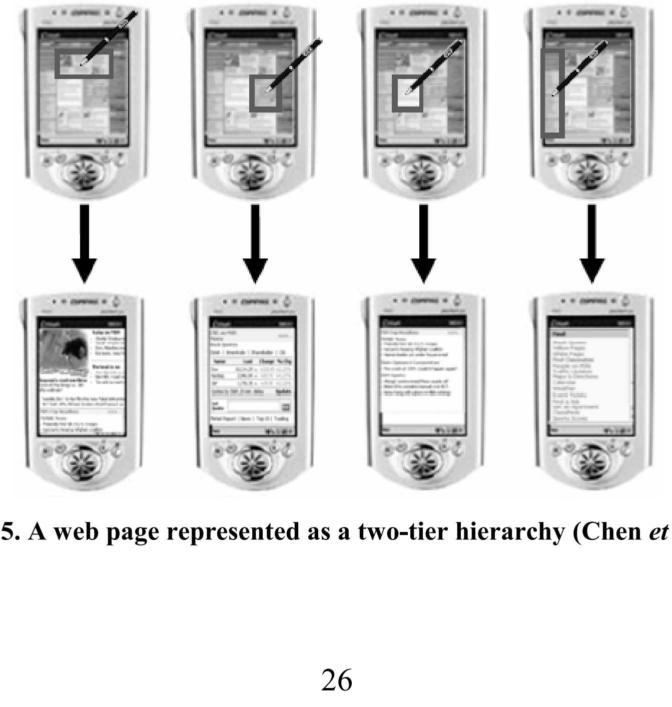


Figure 2-5. A web page represented as a two-tier hierarchy (Chen *et al.* 2003).

In order to produce a two-tiered representation, two main processes have been implemented: page analysis and page splitting or auto positioning. The web page is first parsed into a DOM tree to allow for manipulation. The DOM tree contains the web page content in hierarchical format, and position and dimension information is stored for each node.

Page analysis attempts to extract the semantic structure of the web page. The resulting structure is represented by a group of nodes, called "content blocks", in which each node is a group of content from the original web page. Page analysis is performed in an iterative manner. Initially, the entire web page is considered to be a single content block. The content blocks are divided into smaller blocks after an iteration of the process. Content is grouped by inferring the content structure of a web page embedded by the author. This is derived from high-level content blocks, such as headers, footers and sidebars, and visual separators consisting of explicit and implicit separators. Explicit separators are created using specific HTML tags, while implicit separators are created by leaving blank space. High-level content block detection attempts to place content into five regions: header, footer, left side bar, right side bar, and the body. Thresholds for height and width are used to determine whether a block is too large and needs to be split further. The positional information in the DOM tree is used to sort the content into high-level content blocks, for example, a threshold *n* defines the upper *n* pixels of a web page that constitutes the header, and content falling within this threshold is included in the header. Explicit separators are identified by analyzing the tags. <HR> is used as a horizontal line in a web page, and tags with borders, such as tables (<TABLE>) define borders which are used as separators. Images are classified as explicit separators depending on their width and height values. Implicit separators are blank areas in the web page and are identified by projecting each basic content block along the horizontal and vertical axes. The widest blank area is then selected from each diagram as the position to partition the block.

On small screens web pages will often be too large to be viewed on a single screen of a mobile device. In this case, page splitting is utilized to remove the need for horizontal scrolling. Single-subject splitting divides the page into several smaller pages which are connected by hyperlinks, and is used for pages with a particular topic. Multi-subject splitting divides the page into smaller pages, and generates an index page which is used to navigate. This approach is used to split an entire web site. Auto-positioning is suggested as an alternative to page splitting. The process is similar in that a thumbnail view of the page is generated, displaying the content blocks in different colors. However, when a particular block is selected, instead of splitting the page it is automatically positioned correctly to display the selected content block.

2.2.4 DOM-Based Content Adaptation

Another content adaptation technique focuses on content adaptation as a means of removing "clutter" from web pages, but attempts to maintain the inherent "look and feel" of a web page (Gupta *et al.* 2003; Gupta *et al.* 2004). This technique attempts to solve various problems encountered by the previous solutions, such as destroying web layout and extracting content from pages with multiple content bodies. This approach has implications for visually impaired and mobile device users. Screen readers for the blind do not automatically convert HTML into plain text before reading. The system, called Crunch, benefits these readers by having the ability to produce plain text as output. The system has been implemented as a web proxy server. An HTML parser corrects the HTML of a web page requested by users and then creates a hierarchical Document Object Model (DOM) tree representation of this page. This allows for the tree to be analysed recursively in sections or as a whole by any of the filters according to user settings.

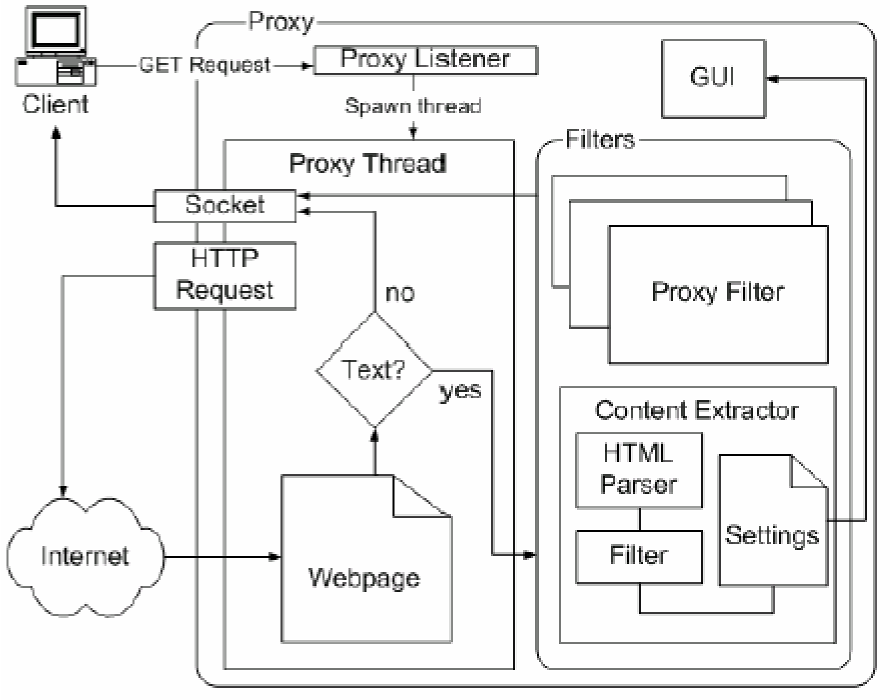


Figure 2-6. Architecture of the Crunch system (Gupta 2005).

There are two types of filters developed in the Crunch application. The first filter ignores tags or attributes within tags, allowing images, links, scripts, etc. to be easily removed. The second, more complex set of filters consists of an advertisement remover, a link list remover, an empty table remover, and a removed link retainer. An advertisement remover identifies advertisement links by comparing the link destinations to a list of known advertisement servers. The server addresses obtained from the links are compared to a table of known advertisement servers and the tags containing these fields are omitted if a match is found. The list is periodically updated from a web site that provides these details. A link list remover identifies and filters large link lists, which often occur on the sides of web pages, leaving the main body of relevant content intact. Gupta (2005) defines link lists as bodies of content where the ratio of the number of links to the number of non-linked words is greater than a pre-defined threshold. If the threshold is exceeded then the contents of the cell are removed. Running these filters may result in completely or partially empty tables. These tables occupy space on the filtered web pages. Tables are examined for user-specified tags that are considered to contain “substance”. A threshold level is set for the amount of relevant content in a table, and the table is removed if this threshold is not met. A removed link retainer was implemented as a backup to provide users access to removed links. All removed text links are stored as they are removed, and added to the bottom of the web page at the end of the filtering process.

Crunch exhibits various limitations. Filtering is restricted to HTML content. Non-HTML content can either be allowed or filtered; it is not possible to filter within the content. Dynamically generated pages, and scripts such as java script, do not filter well and therefore must also be either completely omitted or allowed. The system does not permit personalisation as the settings can only be changed on the proxy and this affects all users.

2.2.5 A Framework for Building Customized Adaptation Proxies

According to Rubinztejn *et al.* (2005), “there is an increasing demand for flexible and extensible tools and frameworks for the rapid development and customization of proxy-based architectures”. This statement can be justified by the large number of proxy-based content adaptation systems developed (Alam & Rahman 2003; Bickmore & Schilit 1997; Gupta 2005; Han *et al.* 1998; Han *et al.* 2000; Lum & Lau 2002). Rubinztejn *et al.* (2005) present a framework for developing application proxies that incorporates the functions common to these adaptation proxies. The proxy framework consists of two sections: the communication sub-system, which handles communications between the clients and servers, and the proxy core, which handles context notifications and applying adaptations. Context notifications provide information regarding the execution context of the requesting device, in terms of the access point used and the quality of the wireless connection. The framework focuses on content adaptation and caching, which are vital issues when developing adaptive applications for mobile devices and wireless networks. A daemon, which runs on mobile devices using the framework, collects context information and transmits it to the proxy. The Context Information Service (CIS), which forms part of the proxy, stores this context information and notifies the proxy of any relevant changes that require altering adaptation settings. The main components of the proxy framework include: handover management, caching management, adaptations, message filtering, protocol translation, context management, and service discovery. Handover management deals with the migration of a client to a new network domain, such as the transfer of a client’s session state. Caching management stores application specific data, messages, and user preferences. Adaptation handles the content adaptation, such as image compression, performed for each client based on context information. Message filtering handles the filtering of data delivered to each client based on the client profile. Protocol translation converts between the protocol used by the proxy server, and the possible protocols of the client network. Context management evaluates context information, and informs relevant sections of the proxy about a change in context. Service discovery identifies new services, users, or data available to the system.

2.2.6 Discussion

The majority of content adaptation implementations focus on manipulating web content for more appropriate display on small-screen mobile devices. Owing to the dynamic nature of documents contained on the web, many implementations attempt to infer the intended structure or layout of documents. The techniques either reduce document dimensions, by removing or resizing rich content such as images, or identify document segments and provide a means of navigating these various sections. Content adaptation approaches should reduce document size and complexity, while minimising the processing cost of document manipulation. Many adaptation architectures implement a middleware approach, where the adaptation system functions as a proxy server, receiving client requests, then retrieving the relevant documents, and finally reformatting these documents before returning the content to the clients. Since content adaptation approaches have focussed on manipulating web content, adaptation implementations have been designed to transform mark-up languages. Proxy servers are widely used as middleware applications to providing content adaptation, and remove the need for content providers to maintain multiple versions of content, as the content can be automatically modified using the middleware application.

2.3. Personalisation of Content Adaptation

Web personalisation refers to the process of customising a web site to the requirements of individual clients, and assists in reducing information overload

(Eirinaki & Vazirgiannis 2003). There are two main personalisation approaches (Ramakrishnan 2000):

1. Content-based techniques search for relevant content by identifying key-words contained within the documents.
2. Collaborative-filtering approaches implement data-mining techniques to identify user patterns and preferences, and make use of profiles in order to customise web pages.

A wide range of hybrid implementations also exist.

2.3.1 Composite Capability/Preference Profiles

The Composite Capability/Preference Profiles (CC/PP) is a system for specifying device capabilities and user preferences. These attributes are used to transform content to suit the device specifications and the user. CC/PP is based on the Resource Description Framework (RDF)[x], which is a metadata description language, and provides an infinitely extensible vocabulary for describing device capabilities and user preferences (W3C 2004). CC/PP does not define how documents should be transformed, but provides a platform for specifying settings (Web Standards Project 2005). A CC/PP profile is a two-level hierarchy, where the profile contains one or more components, and each component contains at least one attribute. A component refers to a capability or preference category, and an attribute describes the specific capability or preference of a specific client. A CC/PP profile is specified within an XML document sent to the server. Figure 2-7 provides a graph illustrating an example CC/PP profile. This example profile contains hardware and software components, where the hardware components detail the device capabilities, such as screen resolution and memory, and the software components detail the capabilities of the software applications, such as whether images can be rendered and the HTML version of the browser.

[x] http://www.w3.org/TR/rdf-primer/

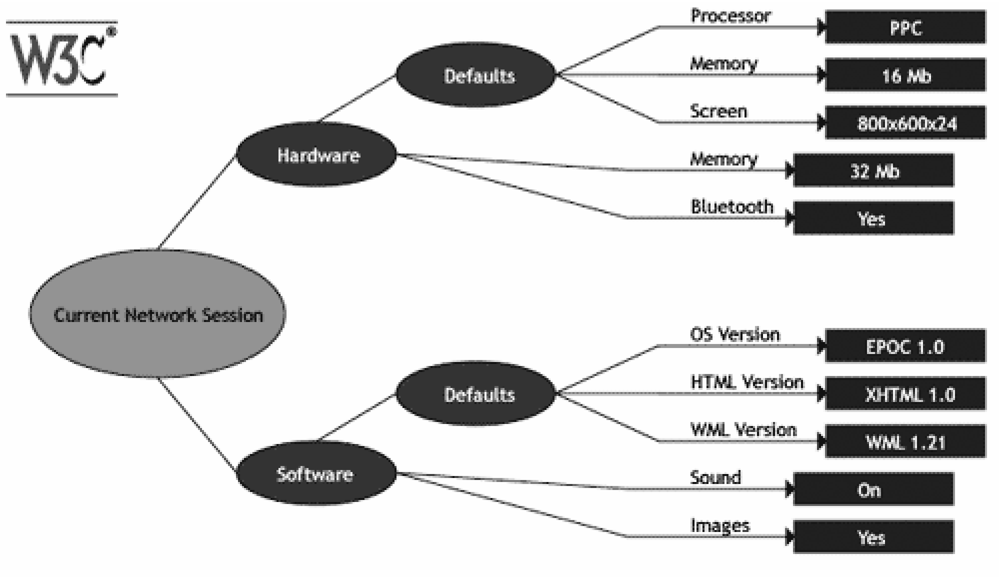


Figure 2-7. An example graph illustrating how the CC/PP can be used to describe device capability and user preference (Web Standards Project 2005).

2.3.2 Web Views

Freire *et al.* (2001) present an architecture for personalising web content and services. The approach allows users to customise personalisation settings, ensuring that the approach is not limited to a set of supported web sites. Since personalisation systems are focussed toward end-users, the approaches should be easy to use. A web view is a simplified version of a web site, as defined by a client. In order to create a web view, the system retrieves the requested document, and then extracts relevant content. The WebView system is implemented as a Java applet, and records the browsing steps of a user. A user creates a web view by simply navigating to a site using the applet, and selecting the content of interest. These generated web views can then be accessed through a server, which is a service on the web that acts as middleware, accepting requests from clients, retrieving the relevant data, and then transforming the document according to the specified web view.

2.3.3 Personalisation by Partial Evaluation

Ramakrishnan (2000) presents a programmatic framework for designing personalisation systems, using partial evaluation, data-mining of semi-structured data, and information integration, known as PIPE. Partial evaluation is an approach to simplify the presentation of a document by removing data that is unnecessary or is not applicable to a certain user. The function takes a web page as input, along with user settings, and creates a modified document based on these preferences. Settings contain key-words which identify desired content. Mining semi-structured data creates a model of a web site by identifying and relating links within the site. The result of this method is a schema describing the web site that can be used for personalising content. Personalisation approaches should integrate information from different web sites, providing cross-referenced content. The PIPE framework requires offline modelling of individual web sites, and is thus limited to a specified domain.

2.3.4 Discussion

Web personalisation approaches attempt to provide a customised browsing experience. Certain implementations attempt to automate personalisation by monitoring user browsing habits and preferences. The CC/PP describes a method for specifying client device capabilities and preferences. Such an approach allows end users to specify their desired settings, which are implemented by a web server. This approach would be best suited to personalising adaptation settings in electronic display networks, as clients are not guaranteed to access the same terminal consistently, making it complex to monitor actions and automate profile development. Personalisation implementations should be easy to use, and require limited user knowledge (Freire *et al.* 2001).

2.4. Adaptation of Content in Response to Bandwidth Limitations

Throughput is defined as the amount of data transferred from one part of a network to another in a certain amount of time, and is measured in kilobits, megabits or gigabits per second. Quality of service (QoS) refers to a level of throughput that is guaranteed by a network operator, and describes the overall experience a user or application will receive over a network (Nortel Networks 2003). This definition can be extended to include the ability of a network element, such as an application or a router, to provide some level of assurance for consistent data delivery services. A variety of technologies, architectures and protocols have been developed to deliver QoS.

Currently, the majority of internet traffic is based on the Internet Protocol (IP). While IP has the benefit of low operational costs, the resulting QoS is unpredictable since packets sent using this protocol do not follow a specific path over a network. The protocol was designed with the focus on reliability, and not on throughput. Network traffic continues to increase in diversity, and thus has different types of bandwidth, delay, loss and availability requirements. IP alone may not be sufficient for certain applications, such as voice applications, but QoS techniques can be incorporated into an IP network to make it capable of providing acceptable, consistent and predictable throughput (Nortel Networks 2003). Until recently, different applications were run on different networks, such as an Integrated Services Digital Network (ISDN) for videoconferencing and an IP network for the internet. QoS technologies were less significant because the traffic on each network was similar, and thus had similar behaviour and requirements. Most applications and networks have converged on being IP-based, and as a result QoS technologies are becoming necessary to provide support for diverse applications in an IP network (Nortel Networks 2003). QoS technologies would not be required if all network connections had sufficient bandwidth such that the network would never become congested. However, this is not the case with most networks today. Even though certain sections of a network may have large amounts of bandwidth, high-bandwidth connections are seldom available from network traffic's source to its destination. These bandwidth discontinuities can create points of congestion on the network and result in unpredictable QoS. In certain circumstances it may be simpler, and more cost effective to add bandwidth as opposed to QoS technologies (Nortel Networks 2003).

QoS can be measured using five parameters; network availability, bandwidth, delay, jitter and loss. Each parameter has an effect on an applications performance, and the experience of a user. Network availability refers to the availability of any portion of a network, and is possibly the most important parameter. Bandwidth is provided in two ways. Available bandwidth is implemented when network operators oversubscribe the bandwidth. All users compete for available bandwidth, and the amount of bandwidth allocated depends on the amount of traffic generated by other users on the network. The guaranteed bandwidth method of allocation, unlike the available bandwidth method, provides a QoS guarantee in terms of minimum bandwidth allocated to each user. Delay describes the time taken for a packet to be transmitted through a network, and issues arise when the delay becomes variable in length. Jitter is the measure of the variation in delay time between consecutive packets. Certain real-time applications are not able to function when the amount of jitter reaches a certain level. QoS can be maintained if the level of jitter is constrained to particular bounds. Loss refers to the actual loss of packets. This can be caused by errors in the physical network, such as fog interfering with wireless transmissions. Alternatively, when nodes in the network become congested, they may begin to drop packets. Some protocols, such as the transmission control protocol, will retransmit dropped packets in order to maintain reliability, but this further congests the network. Two other parameters, emission priority and discard priority, influence QoS but their effects are not easily quantified. For this reason they are used as a method of managing a network, and not to measure the QoS. Emission priority determines which packets should leave a network first based on an assigned priority, and discard priority determines which packets are discarded first during times of network congestion (Nortel Networks 2003).

The majority of work on QoS has focussed on low-level approaches, usually at the network level. However, recently many approaches have investigated influencing QoS parameters by manipulating content at the data level.

2.4.1 Dynamic Distillation

Fox *et al.* (1996) introduce a system that makes use of datatype-specific lossy compression, called on-demand distillation, in order to reduce network latency and increase QoS. The work focuses on reducing issues resulting from differing client devices and differing network capabilities. Clients are defined to differ on three levels: network, hardware, and software. At the network level client networks offer different bandwidth limits and latency. Hardware differences include screen size, processing power and display resolution. Software differences include the data encodings that devices can handle, such as Postscript and JPEG data. Despite the rapid advance in mobile device technology, Fox *et al.* (1996) maintain that the high-end devices, such as laptops, will advance at a similar rate, ensuring that there will always be a divide between the high-end and low-end devices. The system provides a middleware application that allows for dynamic distillation. This approach removes intelligence from both the client and server, and removes the need for pre-processing content. Lossy compression aims to preserve sections of content that have the highest semantic value. For example, by removing formatting from text or reducing the resolution of an image. The main goal of the distillation approach is to allow users to evaluate the importance of data, and then download the original or a section of the original, a process known as refinement. The system architecture illustrates the implementation at the proxy level (Figure 2-8). Client devices communicate with the proxy, which makes use of the distillers to dynamically convert content to reduce latency. The network connection monitor (NCM), monitors the bandwidth resources of the network and informs the proxy of changes. The proxy may respond by changing distiller settings in order to modify the network usage. A user can specify a desired latency value, and the proxy can then request output from the distillers, based on the network conditions, that will provide for this.

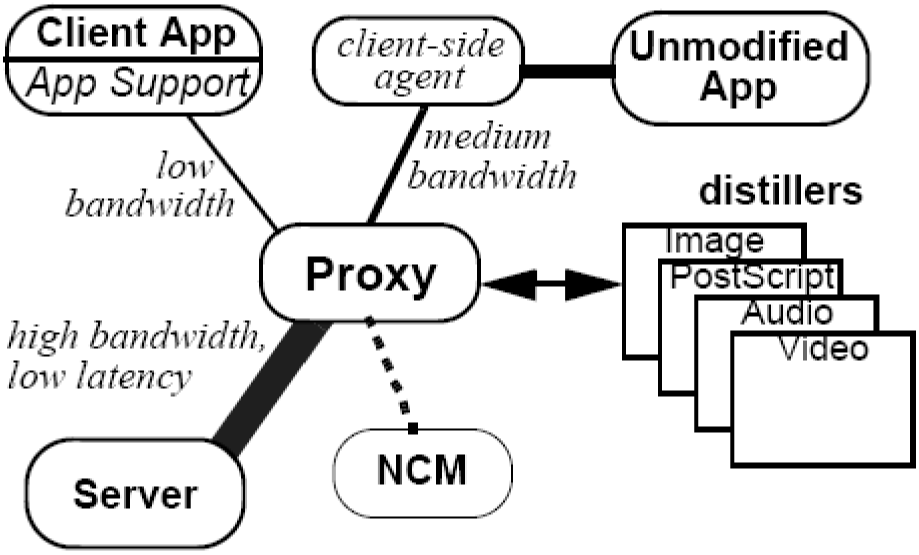


Figure 2-8. System architecture of the dynamic distillation approach (Fox *et al.* 1996).

2.4.2 Web Content Adaptation

Web servers will begin to suffer from a reduction in response time and an increase in connection error rate as the rate of connections exceeds server capacity. Abdelzaher and Bhatti (1999) introduce an approach for avoiding these two errors during network peak usage periods. The system is only effective in networks that have periods of low and high usage, and not in networks that suffer from continuous high load. This approach follows the premise that clients would be prepared to receive slightly degraded content, opposed to the alternative of suffering from connection errors.

Aside from bandwidth constraints, many connection requests are processed before failing, using up server CPU and memory resources. Content is pre-processed in order to improve the efficiency of the system. While there are a large number of possible permutations of processing settings, two main approaches have been implemented. These are the minimal adaptation agent, and the enhanced adaptation server.

The minimal adaptation agent approach selects between two content trees, one tree containing the original content and the other containing degraded content. The load monitor periodically tests the server queue length by sending a request and measuring the response time. Adaptation of content occurs when the response time exceeds the pre-defined maximum length. The enhanced adaptation server monitors the request rate of the server, and adjust the adaptation settings accordingly. These two approaches provide only for switching between two content trees, which causes a large difference in network usage. At the point of adaptation, a large drop in usage will occur, causing the under utilization of available bandwidth (Figure 2-9). In order to solve this issue, the implementation degrades content for only the fraction of the clients required to avoid over utilization of the server (Figure 2-10).

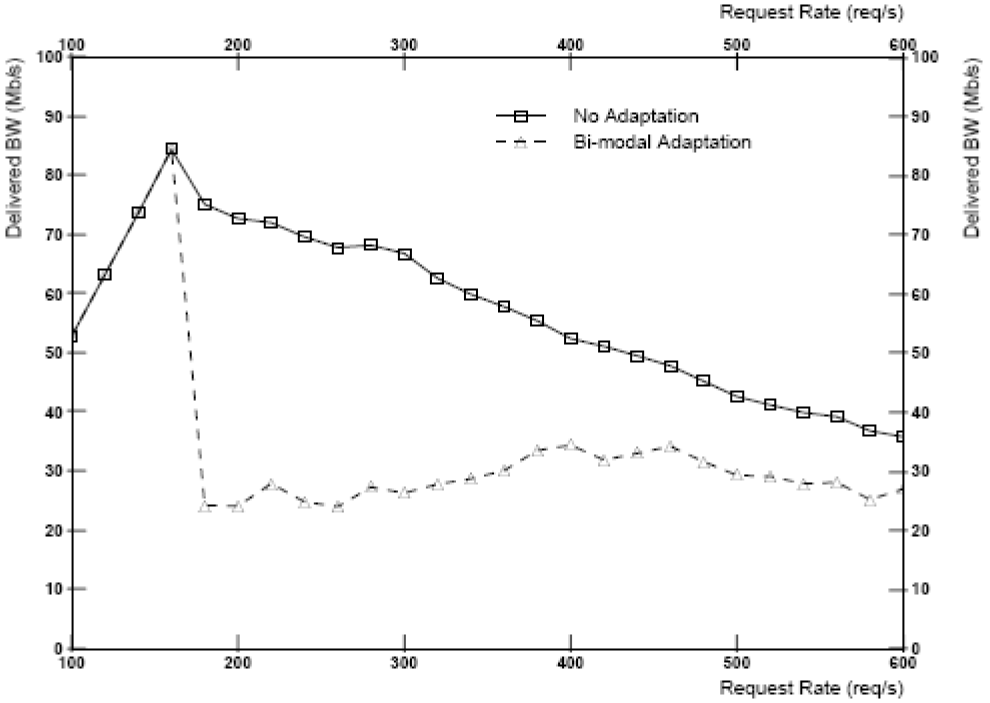


Figure 2-9. Under utilization of network resources (Abdelzaher & Bhatti 1999).

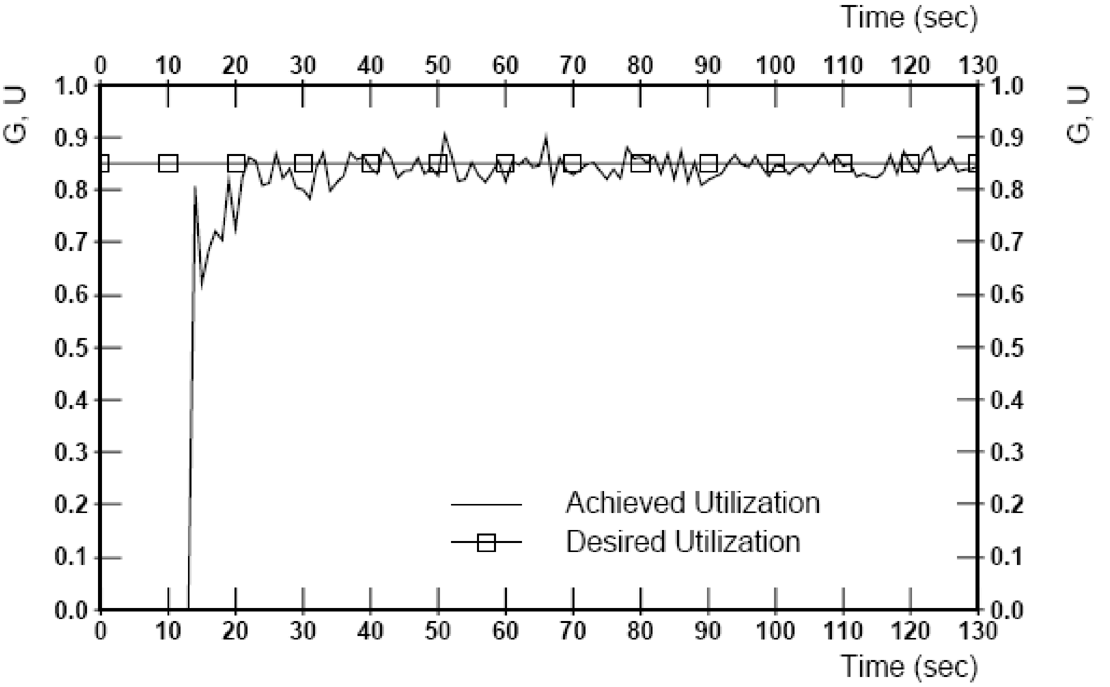


Figure 2-10. Controlled usage of network resources (Abdelzaher & Bhatti 1999).

2.4.3 Context-Aware Decision Engine for Content Adaptation

Lum and Lau (2002) present a system that attempts to solve the disparity between rich multimedia content and the constraints of mobile devices. The approach makes use of a proxy server to perform content adaptation. A user is identified after a request, and their preference settings are applied to the requested content using a decision engine and a content transcoder. Content is transformed according to a content value, which is derived from user preferences. This should provide optimal satisfaction, since QoS can be classified as a user perception. The process of selecting the best content adaptation algorithm is performed by a process known as content negotiation, which comprises two phases: pre-processing and real-time processing. Pre-processing generates a list of all possible score nodes, where a score node is a node in a linked-list or tree representing a version of the content and a corresponding QoS score. Equation 2-1 illustrates an example of how a score representing QoS can be calculated. Each parameter is evaluated using a certain quality criteria, for example, colour depth. Users can rank the parameters according to personal preference, so one may be considered more important than another. Each parameter is then weighted, and the aggregate for all weights calculated. This value is the main input for the decision engine and is used to identify the ideal node.

$$QoS_{pdf-document} = (\text{color, scaling, segment, downloading-time, modality})$$

Equation 2-1. The equation used to quantify a content document, in this case a PDF file.

Real-time processing selects the appropriate node and transforms the content according to the settings stored within the node. Each score node is examined and evaluated according to the adaptation settings in the score node, the actual content, the network conditions, and the capabilities of the client device. The result of this process is either True or False (Equation 2-2). A result of True means that if the client is sent the requested content, transformed according to the settings in the score node, the client device will successfully render the content. Search algorithms are provided to improve the search time through the linked list or tree.

$$T \,||\, F \longleftarrow \text{decision(score-node, content-metadata, network-parameters, device-capability)}$$

Equation 2-2. Binary decision used to select the optimal node.

This approach has been implemented as a system to convert PDF[xi] documents for display on mobile devices. Documents can be adapted according to colour, size, segment, downloading time, and modality. The PDF documents can be converted into WML, HTML, Bitmap, and PDF format. User willingness to allow this is specified by the modality parameter. Device parameters are also considered, including screen size, colour capabilities, supported media types, supported mark-up languages, and the memory buffer size. Network parameters evaluated are available bandwidth and round-trip time. User IDs are embedded in URLs in order to identify them to the system. The system successfully adapts content in response to changing network conditions, device capabilities, and user preferences such as minimum downloading time.

[xi] The Portable Document Format (PDF). For a description see http://en.wikipedia.org/wiki/PDF

2.4.4 Remote Execution

Newhouse and Pasquale (2005) present a system for remote execution of code, known as Java Active Extensions. These extension systems allow either a client or server application to load and execute code. This reduces the load on the client or server, freeing resources. There are three possible extension models: client-extended, server-extended, and private server. Client extensions can filter and cache data, bridge protocols, monitor and react to network conditions, or provide an anonymous connection. Server extensions can increase scalability by distributing functionality throughout the network. These two models facilitate interaction between two endpoints, whereas the private server model interacts with only a single endpoint to provide additional resources. The extension model is illustrated in (Figure 2-11). Four steps are required for an endpoint to launch an extension:

1. Discovery – In order to make use of extensions the endpoints must be aware of the location of available extension systems. Extensions register with a directory service, which can then be accessed by endpoints.
2. Resource allocation – Endpoints can request extra bandwidth if required, the extension ensures that each endpoint process consumes only the allocated resources.
3. Extension deployment – Endpoints supply extensions with a URL pointing to the code required to be run. The extension retrieves this code.
4. Execution – The extension executes the retrieved code, controlling the resource usage as required.

Potential benefits include increased performance, reliability, and reduced power usage for mobile devices. This improves processor QoS. The operating cost consists of the messages required to launch an extension.

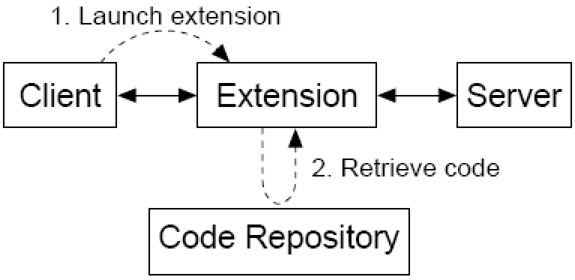


Figure 2-11. The Java Active Extension model of remote execution.

2.4.5 Discussion

Content adaptation techniques attempt to reduce document size and complexity, and thus inherently reduce network load. Network management implementations, utilising content adaptation approaches, monitor network conditions and respond to increases in network load by applying adaptation techniques. Abdelzaher and Bhatti (1999) describe a network underutilisation problem caused by these network management approaches. Underutilisation is caused when adaptation techniques are applied, causing a large drop in network usage, resulting in unused bandwidth. A possible solution to this issue is to transform a fraction of documents processed by the server.

2.5. Summary

A simplified electronic display system consists of a display terminal, which incorporates a controlling computer and a large screen display. The widespread installation of electronic display networks provides the opportunity to expose the displayed information to mobile device users. However, current implementations provide, at best, limited access to content where specified documents are transferred to the requesting device. Content adaptation techniques attempt to solve the issues introduced by accessing web content using small-screen devices. Thus adaptation techniques have been designed for use with mark-up languages such as XML and HTML. Owing to the wide variety of document layouts on the internet, certain adaptation approaches attempt to automatically infer document structure. While these approaches allow for a wider range of documents to be browsed, the computational cost of adaptation is significantly increased. Adaptation architectures make use of a middleware design, functioning as a proxy server. This approach ensures that all requests are handled by the adaptation system, and content providers need not maintain multiple versions of content. Personalisation systems provide customised web pages for specific clients, based on device capabilities and user preferences. Automated personalisation implementations monitor user behaviour and automatically adjust content accordingly. The CC/PP provides a system whereby users can specify device capabilities and preferences using an extensible XML profile. These settings can then be used by a proxy server to adapt content accordingly. Content adaptation can be used as a method of network management to reduce network load. These systems monitor network traffic and implement adaptation techniques as required. Such approaches may cause an underutilisation problem, where the adaptation techniques reduce network load to the point where available bandwidth is not utilised. The requirements and shortcomings of electronic display networks, content adaptation, personalisation, and responsive adaptation, identified in this Chapter, will be used in the next Chapter as a basis for developing an adaptation framework for electronic display networks.

Chapter 3

PROPOSED FRAMEWORK

Chapter 2 examined related work in the fields of electronic display networks, content adaptation, content personalisation, and adaptation network management strategies. The outcomes of this examination provide the basis for developing a framework to extend electronic display systems to allow wireless access, and to further facilitate this access by implementing content adaptation techniques. This chapter describes a generalised model for existing electronic display systems, and provides a broad overview of a proposed framework for extending these existing implementations. Both the physical architecture and the generalised object model of the framework are presented. The design of the framework is based on the requirements established in the previous Chapter.

3.1. Existing Architectures

Numerous electronic display systems exist, and these systems are generally designed to convey information to observers in a digital, multimedia environment. Figure 3-1 illustrates a generalised overview of an electronic display system, based on the electronic display systems described by Harrison and Andrusiewicz (2004), and the Omnivex Corporation (2005).

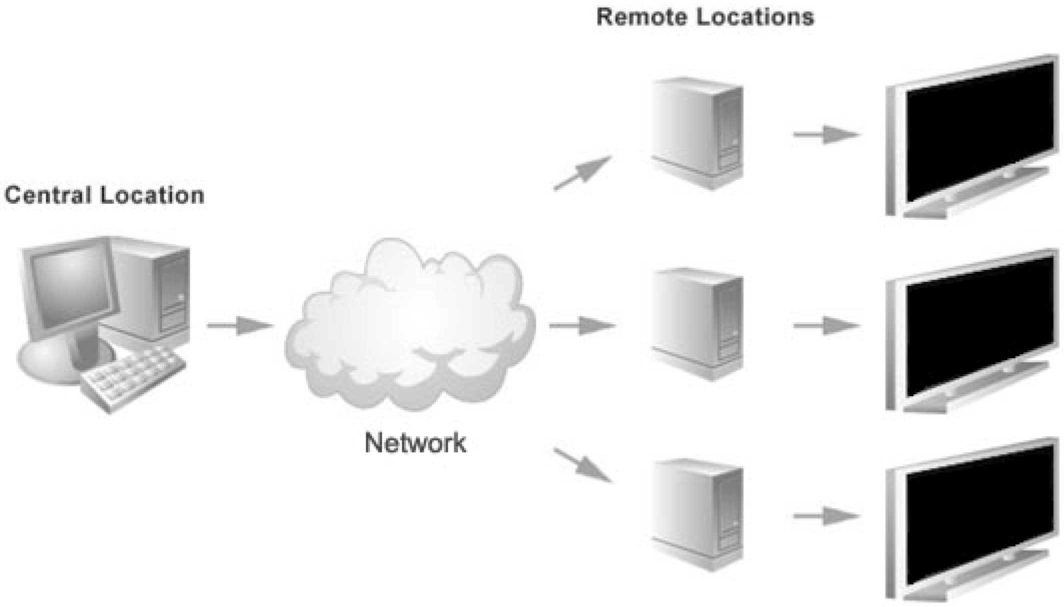


Figure 3-1. Generalised overview of an electronic display system (Omnivex Corporation 2005).

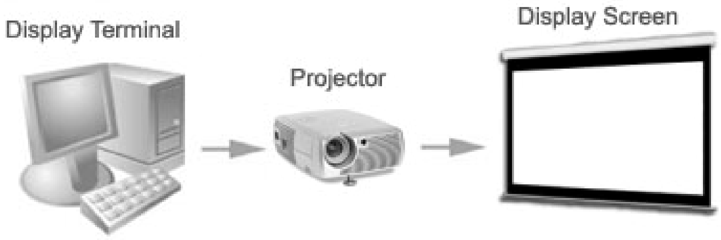


Figure 3-2. An autonomous display terminal. This example shows a computer displaying content on a screen using a projector.

A display terminal consists of a controlling computer, and display mechanisms such as a plasma screen or projector (Figure 3-2). The associated computer controls the content on the display medium. This content may be stored locally or on a remote content server, and may incorporate all types of multimedia content from text to live video. Display terminals may exist autonomously, or can be connected to form a network. When connected, these terminals are usually controlled from a central server (Figure 3-1). The control server will send display details to relevant terminals at the appropriate times. This can be generalised to be a single display terminal, since any one terminal can be replicated and added to the system, making the architecture scalable. A short-coming of this approach is that access to information is limited to viewing of the display screen (Harrison & Andrusiewicz 2004). Observers are not able to access data beyond viewing the screens. Harrison and Andrusiewicz (2004) have developed a system where content is delivered to mobile devices on request. However, this approach allows only a predefined set of content objects to be transmitted when requested, and might not provide access to the desired content.

3.2. Goals of the Framework

The aim is to develop a framework that can extend the generalised, existing electronic display system (Figure 3-1) to provide deeper accessibility to the content shown by the display terminals in the network. This would allow mobile device users access to the information displayed on electronic display systems, permitting them to freely browse any and all content. More specifically, the framework should provide:

- Wireless access - This will provide convenience, allowing client access using mobile devices. The type of wireless access is insignificant, for example 802.11g or 3G technologies.
- Content adaptation - The adaptation techniques will display content in a more convenient manner for the mobile devices, assisting to address issues such as limited screen size and processing power. Lum and Lau (2002) state that "the mismatch between rich multimedia content and constrained client capability presents a research challenge". The main goal of content adaptation techniques is to reduce document dimension and complexity. Content adaptation techniques may also positively influence network metrics and billing models, by reducing the amount of content delivered across a network.
- Modularity and extensibility - The framework should provide for the addition, and/or replacement of system objects. Modularity allows sections of the framework to be replaced seamlessly, replacing a particular content adaptation module for example. Modularity should also apply to the hardware technologies. The system should continue to function if the networking technology or the type of display screens used is altered.
- Remove complexity from clients and servers - This is achieved by implementing the logic as a middleware application. This removes the need for servers to maintain multiple versions of the same content (Fox *et al.* 1996). Servers also need not perform any additional processing which may slow

down round-trip time. The middleware application should exist close to the network clients, as moving functionality to the clients can reduce latency in a network (Newhouse & Pasquale 2005). This approach reduces load, in terms of processing and memory use, on the potentially limited client devices.

- A generalised, flexible framework - The framework should be applicable to the majority of existing electronic display system architectures. This would increase the domain of the framework, making it more applicable to a greater variety of applications.
- Personalisation - The system should attempt to handle personalisation of content adaptation settings. Such an approach would allow users to define their own adaptation settings, based on their device capabilities and personal preferences, thereby improving the user experience. Personalisation approaches should be easy to use, requiring only limited client knowledge.
- Responding to network conditions – The approach should have the ability to respond to changes in network conditions, by modifying the adaptation settings in order to reduce the amount of network traffic. This would free resources, such as bandwidth, allowing more users to access the network while maintaining a relatively constant level of QoS. Abdelzaher and Bhatti (1999) describe a network underutilisation artefact caused by adaptation network management implementations. The responsive adaptation approach should implement techniques to address this issue.
- Low response time overhead – The adaptation modules and framework extensions should not detract from the user experience by significantly increasing the response time for users. Thus the performance of the modules and extensions is an important aspect of the framework, and the processing time required to perform framework tasks should be minimised.

3.3. An Adaptive Extension to Existing Architectures

The extension to the architecture is designed to meet the goals described in Section 3.2. The generalised electronic display framework (Section 3.1) is extended by adding a wireless access point to each computer associated with a display terminal, coupled with a software application running on every computer (Figure 3-3). Note that the same can be done for autonomous display terminals, such as the one shown in Figure 3-2. The access point allows mobile devices to connect and access data stored either locally on the display terminal or remotely on another content server. The software application controls device access, and handling of requests and responses much like a proxy server. This software also provides content adaptation functionality, and can be extended to provide further features, such as personalisation of adaptation settings. Placing the access point at the location of the display terminal greatly simplifies location issues in terms of locating users, and relevant content is already identified.

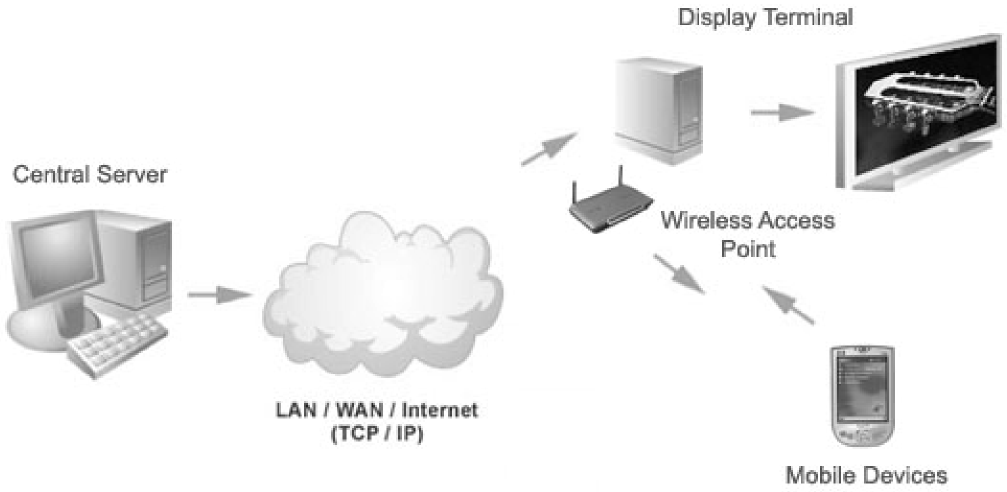


Figure 3-3. Extension to digital signage systems providing wireless access.

3.3.1 System Architecture

The software application is the main focus of this research. As mentioned previously, the software is designed to be modular. Figure 3-4 provides an overview of the architecture of the adaptation system. A client, using a mobile device, requests a content document. The proxy server receives this request and retrieves the relevant document, which is then passed through all installed adaptation modules before being returned to the client. A personalisation approach can be implemented to extend the framework. Here, adaptation preferences are stored on the client device, and are transmitted to the proxy server which stores the settings in a user settings database. When the user requests a document, these specific preferences are applied. The network monitor provides the adaptation modules with network statistics, which can be used to alter adaptation settings in order to control the conditions of the network.

The architecture is similar to the approaches described by Gupta *et al.* (2004) (Chapter 2, Section 2.2.4), and Rubinztejn *et al.* (2005) (Chapter 2, Section 2.2.5), also making use of a middleware approach by placing a proxy server between the content documents and the client devices in order to facilitate adaptation.

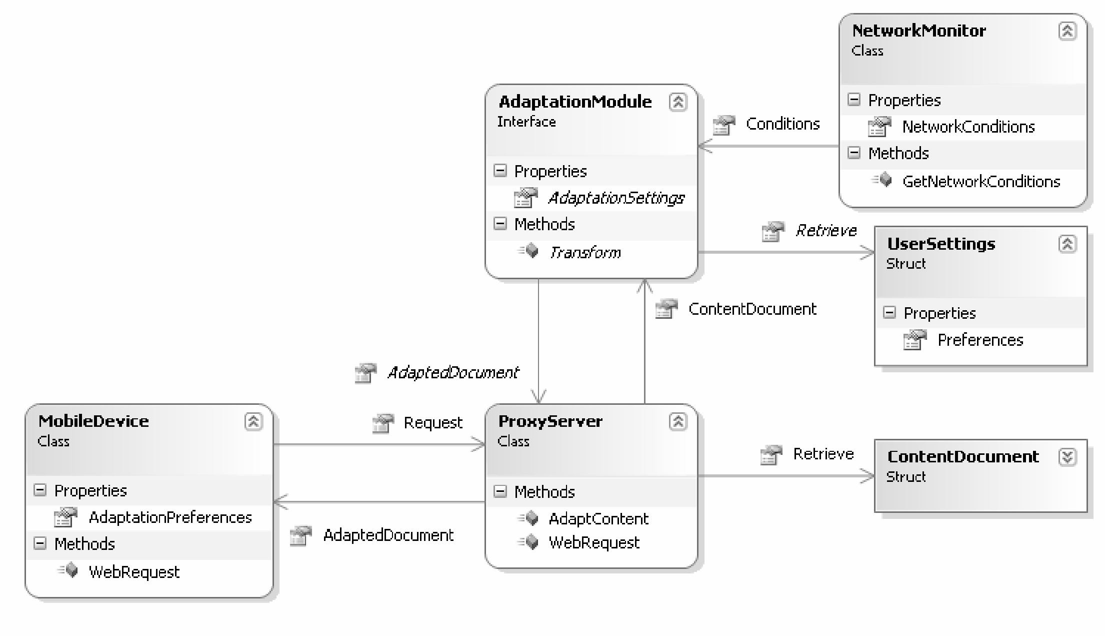


Figure 3-4. Architecture of the software application of the adaptation system.

3.3.1.1 Content Document

A content document describes a content display shown on an electronic display system. The documents are stored either locally on the display terminal computer, or at a remote location. For the purposes of this project content documents are assumed to be written using a mark-up language, such as XML, but the approach could be modified to handle any document format. The content adaptation framework utilises a proxy server to transform content documents by using content adaptation techniques. Adaptation approaches have typically been designed for manipulating web content, owing to the inherent structure in mark-up languages. In order to facilitate content adaptation at the proxy server and to assist in removing complexity from the clients and servers, the framework is designed for use with HTML and XML documents.

3.3.1.2 Mobile Device

The mobile device is the client device and can have a wide range of capabilities. In order to access and view content documents, the client should only require a web browser. This assists in removing complexity from the client-side. An optional extra in this architecture is to install some functionality on the client device that stores the user's adaptation preferences, and transmits these along with the document request. It would be possible to remove the need for any client-side software by implementing the setting of user preferences as a web application. This would provide the same functionality, but would keep the complexity on the proxy server.

3.3.1.3 Proxy Server

The proxy server acts as a middleware application between client devices and content documents. Implementing the system as a proxy removes complexity from content servers and clients, and removes the need for content servers to maintain multiple versions of content documents as the content is adapted automatically. Client requests are intercepted and the relevant document is retrieved. At the same time any adaptation preference information sent by the client is identified and stored as user settings. Once the content document has been retrieved it is first sent to the adaptation modules before being returned to the client. The proxy server could facilitate other functionality, such as billing clients for network usage and client authentication. Adaptation techniques and framework extensions can be added or removed from the proxy server, providing modularity and extensibility.

3.3.1.4 User Settings

A database is utilised to store the user settings, for a single session, for each client. The preferences are selected by users on their devices, and transferred to the proxy that then stores the settings in the database. Adaptation settings may include a wide range of choices, including device capabilities and content settings such as resizing images and removing formatting from text. The personalisation module is an optional addition to the framework, which adds further functionality but is not an integral part thereof. While the personalisation extension facilitates the improvement of the user experience by allowing customisation, it may require the installation of client-side software, which may detract from the goal of reducing client side complexity.

3.3.1.5 Network Monitor

The network monitor polls the network, at a set time interval, in order to examine the conditions of the network. Network conditions may include (but are not limited to): bandwidth usage, packet round-trip time, length of the server request queue, latency, and connection errors. When the network metrics reach a pre-specified level of usage, the network monitor notifies the adaptation modules. The network monitor module is an optional section of the framework that provides for the implementation of a network management module.

3.3.1.6 Adaptation Module

Adaptation modules receive a content document as input from the proxy server. The document is then parsed to identify the content objects contained within the document. A variety of content adaptation techniques can be applied to the document before it is returned to the proxy server. Methods of content adaptation may depend on information obtained from the network monitor or the user settings database. Modules are implemented as plug-ins, allowing additional adaptation modules to be easily added to the system. The adaptation modules provide a means to reformat content documents to display appropriately on small-screen mobile devices, and facilitate network management by providing a method to reduce the size of content documents.

3.4. Summary

Electronic display systems consist primarily of display terminals, which are computers controlling large-screen devices. The main requirements for a content adaptation framework include: wireless access for mobile devices, content adaptation facilities, a modular and extensible framework, remove complexity from clients and servers, a personalisation approach, response to network conditions, and a low response time overhead for all modules and extensions. The proposed framework applies to each display terminal in an electronic display network. A proxy server is installed on the terminal, and provides wireless access to content through a wireless access point. The proxy server also provides functionality to install content adaptation modules, to provide a method to reformat content for small-screen devices. This removes the need for content servers to store multiple copies of content, thereby removing complexity from the servers. A module, residing on a client's mobile device, provides users with the ability to specify specific content adaptation preferences. These settings are sent to the proxy server, which stores the preferences as session information in a local database. Content is then adapted according to the preferences of individual clients. A network monitoring extension maintains network statistics, which can be used to manipulate content adaptation settings in order to reduce the load on the network. Chapter 4 details the implementation of three content adaptation modules, a personalisation extension, and a network management extension, which will be used to extend existing electronic display systems in order to evaluate the framework.

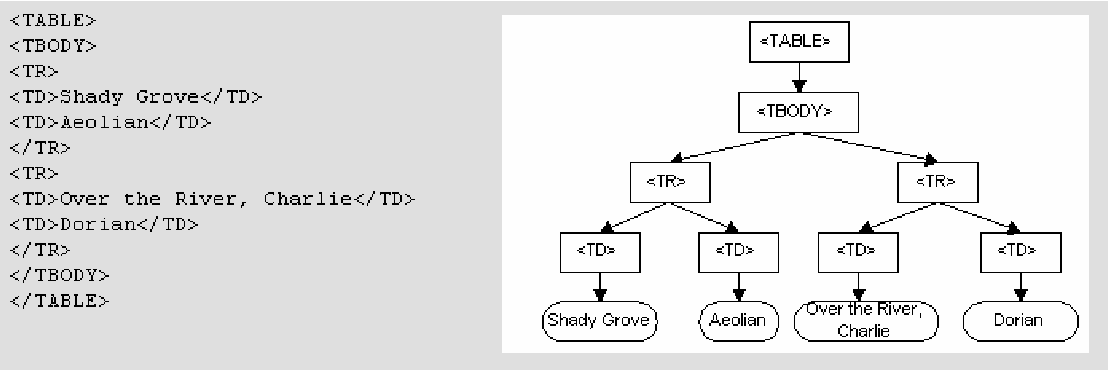
Chapter 4

FRAMEWORK MODULES

Chapter 3 described a proposed framework for extending electronic display systems to provide mobile device access. The framework consists of a proxy server, content adaptation modules, a personalisation extension, and a network management extension. Numerous proxy server implementations exist, and so it is not necessary to develop a new proxy system. This project makes use of the Crunch proxy server, which is an open source thread-based proxy written in Java, and provides for the implementation of adaptation plug-ins (Gupta *et al.* 2004), and Microsoft ASP.NET, which is a web server application that allows for the development of mobile web applications and provides advantages such as caching and device filters. Three content adaptation modules have been implemented. Each module takes a different approach to adaptation, and the three implementations will be used for comparative purposes. The aim of the adaptation modules is to reformat content documents to display more appropriately on small-screen mobile devices. A personalisation system has been developed that allows users to specify their desired adaptation preferences, for any adaptation module. The aim of this personalisation module is to provide a means for users to customise content documents in an easy to use environment. A network management facility has been implemented, and can provide network statistics to any requesting adaptation module. This approach aims to monitor network usage, and adjust adaptation settings in order to maintain a specified QoS level. The content adaptation modules, personalisation extension, and network management extension, described in this chapter, will be used to extend existing electronic display systems according to the proposed framework. In order to emphasise the modularity of the framework, different programming languages where used to implement the adaptation modules.

4.1. Content Adaptation

Content adaptation has traditionally been used to transform web pages in order to display the content more conveniently on the small screens of handheld devices. XML and HTML function correctly in this regard owing to the inherent structure in mark-up languages. There are two types of XML parsers: Event-based and Tree-based. Event-based parsers, such as the simple application programming interface for XML (SAX), reports events directly to the application using call back methods. Tree-based parsers, such as the document object model (DOM), read an entire XML document into a tree structure in memory, and the application can then navigate the tree. SAX parsers are efficient in terms of CPU and memory usage, but DOM parsers are better suited to manipulating XML documents. Parsers are available, such as the Xerces[xii] Java Parser, which can convert XML and HTML documents into DOM representations, and can then be traversed and certain nodes modified or removed. The DOM defines the logical structure of a document and the way it is accessed and manipulated. Figure 4-1 shows an example table taken from an HTML document, and a graphical representation of the resulting DOM tree representation. HTML and XML documents can be built, navigated, and elements and content can be added, deleted or modified. The DOM is a language independent specification. The majority of parsers create a tree representation when converting to a DOM (W3C 2000). It is also not necessary to discriminate between XML and HTML, and thus both languages can be employed. Owing to these advantages, and the related work already carried out with content adaptation techniques on HTML (Section 2.2.2, Section 2.2.3, and Section 2.2.4), mark-up languages were chosen as the document description language. The DOM specification is standardised, and thus it is possible to switch parsers as desired without affecting the results of the adaptation modules. This could allow faster parsers to be incorporated into the system.



```
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```

Figure 4-1. An HTML table, and the corresponding DOM tree, represented graphically (W3C 2000).

[xii] http://xerces.apache.org/xerces-j/

Three content adaptation techniques have been implemented, and the algorithms are described below. The content adaptation implementations aim to reduce document dimension, size (in bytes), and complexity, resulting in documents that display more appropriately on small-screen devices. Tag extraction provides a method of manipulating the content within documents, allowing the dimension and size to be reduced. Text summarisation and spatial representation divide documents into segments, providing a means to navigate between the different sections. The main goal of these two implementations is to provide a convenient means of browsing documents on small-screen devices. All experiments carried out were aimed at establishing the response time overhead associated with each approach, and to evaluate the correctness of the approach in producing the expected result. Usability of the modules was considered to be out of the scope of the project and was thus not a consideration in evaluating the adaptation modules.

4.1.1 Tag Extraction

Content documents often contain a variety of multimedia objects, such as images, audio, and video. In some cases it is possible that a user accesses a page in order to retrieve information in a low-level format, such as text, and is not concerned with images. The tag extraction process works on the principle that certain content is not necessary, and can be used to remove, replace, or modify certain content tags within an HTML or XML document. Objects can be completely removed, for example removing images, or altered to reduce complexity, such as removing formatting from text. The main goal is to reduce the complexity of a content page, but the approach also has the ability to reduce the size of the document in terms of both dimension and number of bytes.

The tag extraction module was written using the Java 2 Standard Edition (J2SE), and makes use of the Java API for XML Processing (JAXP)[xiii] which provides XML parsing and manipulation functions. In this tag extraction approach, a DOM tree representation of the document is created, and is then parsed to identify pre-defined tags by matching a pre-defined list of tag names to the nodes in the tree. These tags

[xiii] http://java.sun.com/webservices/jaxp/index.jsp

are then replaced with a link to the original content. Tags that require extraction are explicitly specified in the tag extraction settings, for example specifying <IMG> will result in all the images in requested documents being replaced by links. Figure 4-2a shows a DOM tree representation of an example HTML document containing some text and a single image element. The tag extraction process traverses the tree, looking for matches to the <IMG> tag. When a match is found, the tag is replaced by a link tag, <A>, which points to the image location. The resulting DOM tree representation is shown in Figure 4-2b. Once the module has traversed the entire tree, the adapted tree is returned to the calling process. Tag extraction settings are altered in code, as no graphical user interface was developed.

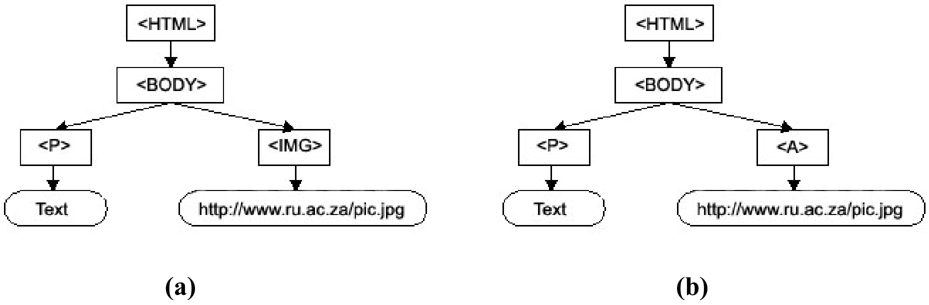
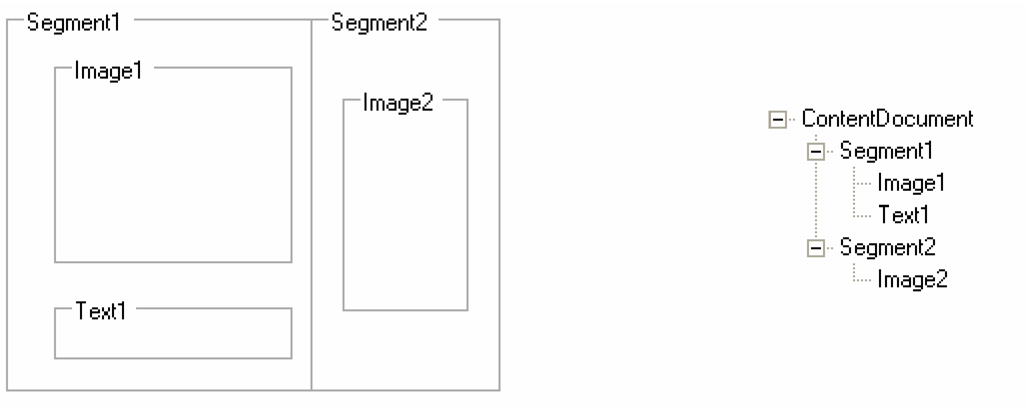


Figure 4-2. The tag extraction process. (a) The HTML DOM tree in original form. (b) The HTML DOM tree after the tag extraction process, note that the <IMG> tag has been replaced by a <A> tag.

4.1.2 Text Summarisation

The majority of content documents are designed to be viewed on large screen devices, such as desktop computers. Text summarisation divides a document into sections, providing navigation through a tree view structure. This allows content documents to be more conveniently displayed on small screen devices, as the dimensions of the documents are reduced. The tree view object provides a convenient method of navigating a document, representing the content in a structured manor. In order to create an accurate representation, where the tree corresponds intuitively to the content document, it is necessary to either have the capability to correctly derive structure from a variety of documents, or to work in a domain of similarly structured documents.

A text summarisation module was developed using C# in the .NET platform. An XML document is parsed using the `XMLTextReader` class[xiv], which is the standard .NET XML parser. `XMLTextReader` is a forward only parser, designed to provide fast access to XML data. In order to avoid complexity, we make the assumption that all content documents analysed by the summarisation algorithm will be within the same structural domain. The module creates a tree view object two levels deep, where the first level corresponds to a content segment and the second level corresponds to all the content objects contained within the parent segment. In this way tree nodes are created simply by identifying tag names and associating the node with the corresponding content. A pre-specified list allows tag names to be identified as document segments, which are added to the tree view as first level child nodes. Any subsequent content nodes within these segment nodes are added as child nodes for that segment. Figure 4-3a shows an example content document, consisting of two segments and three content objects. Figure 4-3b shows the corresponding tree view object, providing a link to each segment and each content object of the document. A user selecting a tree node is provided with only the XML content contained within that node, including all child nodes. It is possible to adapt the approach to different content documents by changing the tag names that are used to identify tree nodes. The module provides only functionality for navigating structured documents.



(a) **(b)**

Figure 4-3. (a) Outline of a content document showing the content segments and content objects. (b) The corresponding tree view representation.

[xiv] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html /frlrfsystemxmlxmltextreaderclasstopic.asp

Tree view objects are created using the Tree View for ASP.NET[xv] control developed by PowerUp ASP[xvi]. The control was chosen owing to the rich feature list, specifically:

- Dynamic, in-code generation of tree view structures, allowing tree view creation to be automated.
- Client and server events for handling drag and select events, used for programming the navigation rules.
- CSS control over the appearance of the nodes.
- Scroll bars for rendering large tree views.
- Numerous other features exist[xvii], making the functionality of the tree view module extensible.

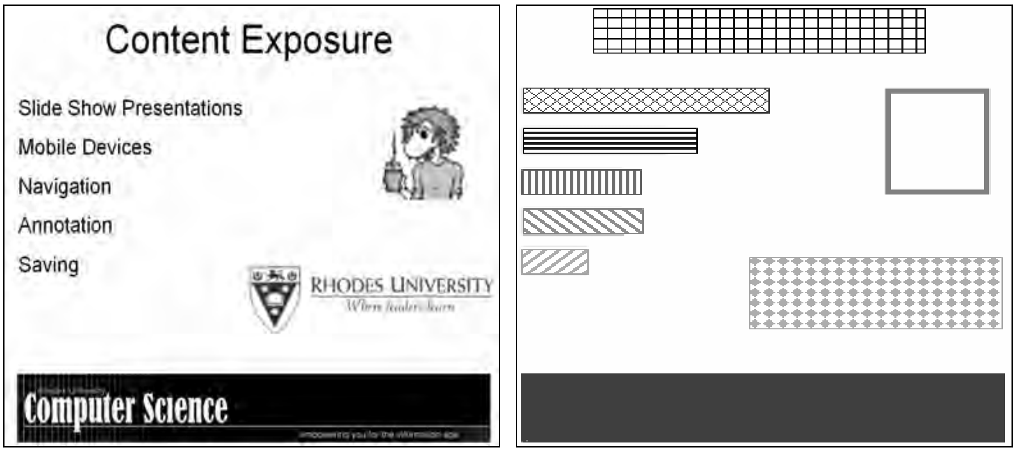
Although the PowerUp ASP tree view control provides functionality to populate a tree view directly from an XML document, the module populates the tree in code. Direct population would require the creation of a further XML document. By dynamically populating the tree in code, the module removes the need for the creation of a further XML file, allowing the tree view to be created using the original content document.

4.1.3 Spatial Representation

Spatial representation is a content adaptation approach that divides up large content documents, allowing the content to be navigated and displayed in a convenient way on small screen devices. The approach is similar to that of text summarization, but provides a graphic representation of the document and the objects contained therein. A representation of the entire document is scaled for viewing on a small screen, and provides links to each content segment illustrated. Figure 4-4 provides an illustration of the approach. Figure 4-4a shows an example content document, consisting of images and text. Figure 4-4b shows a spatial representation of the document, where each block corresponds to a content object depicted in Figure 4-4a. Each block provides a link to the appropriate content, allowing for navigation. The spatial representation can be scaled without losing quality since it is a simple image. In the

[xv] http://www.powerupasp.net/asp.net/treeview/
[xvi] http://www.powerupasp.net/
[xvii] http://www.powerupasp.net/asp.net/treeview/features.aspx

example, a block corresponds to a single content object. However, in practice, a block could be used to represent a content group consisting of a variety of objects.



Content Exposure

Slide Show Presentations

Mobile Devices

Navigation

Annotation

Saving

(a) **(b)**

Figure 4-4. (a) An example content document. (b) A spatial representation of the document.

The spatial representation module implemented is based on the approach described by Chen *et al.* (2003). This module is designed for use with electronic displays, while Chen *et al.* (2003) implement an approach for generalized web browsing. The module represents content pages as a thumbnail image divided into differently coloured content blocks. The approach implemented by Chen *et al.* (2003) creates a thumbnail image of a web page, showing the actual content. This is unnecessary when using the approach with electronic display systems as the actual content is viewable on the large display screen. The module was implemented in Java using the J2SE. XML documents are parsed into a DOM tree using JAXP. The image representation is created by identifying and rendering relevant nodes in the tree using the ICEBrowser SDK for Java[xviii], developed by ICESoft[xix]. The ICEBrowser SDK is a class library that allows web pages, expressed as DOM trees, to be manipulated and rendered. Once the nodes have been rendered, it is possible to obtain the coordinates of the bound box in which each node is contained. This node represents a segment of the document, and thus the coordinates define the location of the segment within the context of the document. These coordinates are used to create an image file containing each segment drawn in a unique colour. The image, along with the coordinates, is then used to create a standard HTML image map of the content page. Each block links

[xviii] http://www.icesoft.com/products/icebrowser.html

[xix] http://www.icesoft.com/

to the corresponding content group contained in the DOM tree. Selecting a block will result in the browser navigating to the appropriate content. The image and coordinates are then scaled according to the screen size of the requesting device. Chen *et al.* (2003) implement a page analysis approach in order to evaluate the intended structure of a web page and identify and group associated content objects (Chapter 2, Section 2.2.3). In order to minimize the complexity of the algorithm, this spatial representation module is designed for a limited domain of XML documents with a restricted structure. Content groups are identified using tag names, as with the text summarization approach.

4.2. Personalization of Adaptation Settings

Automatically adapting content according to device capabilities may not always produce the best user experience. For example, rescaling images to reduce content size may be an effective approach, but certain users might prefer receiving full size images despite the increased download time and strain on their devices. In this case it is impossible to infer user preference without providing a mechanism for the clients to specify their desired adaptation settings. Functionality is provided for users to select their own adaptation settings, supporting users who wish to control the type of content they receive. Figure 4-5 shows an overview of the system architecture of the personalization module. The Browser Helper Object (BHO) intercepts requests from the web browser and appends user preferences before forwarding the request. The proxy server removes the settings and stores them in a SQL database. The adaptation modules access these preferences from the database when handling requests from clients.

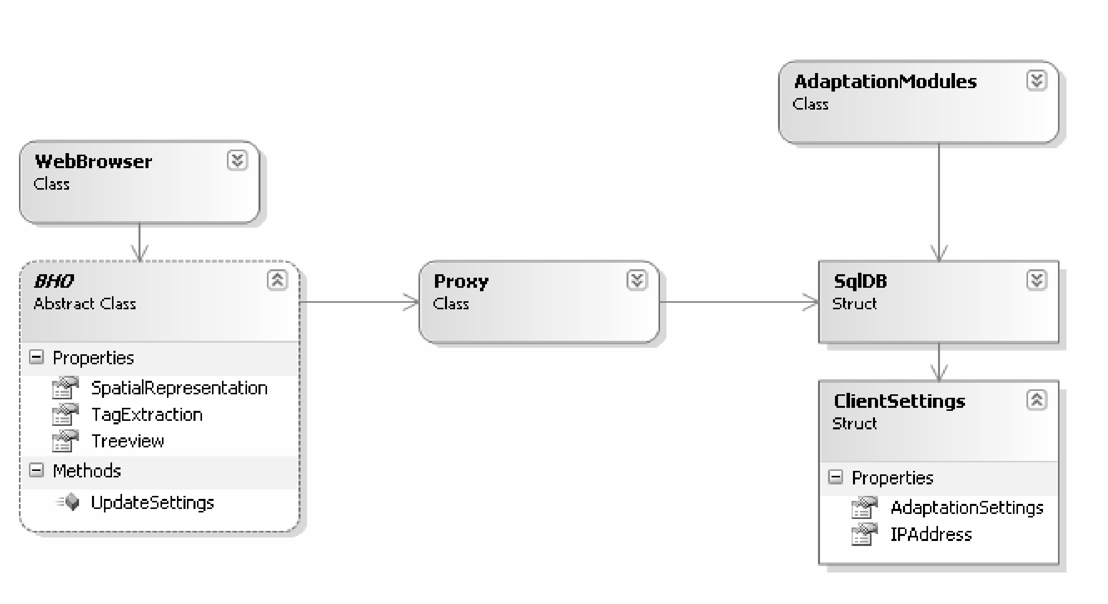


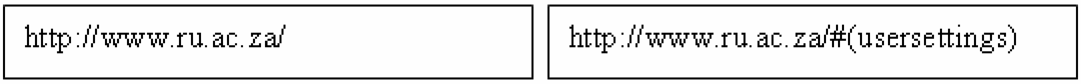
Figure 4-5. System architecture of the content adaptation personalisation approach.

4.2.1 Web Browser and Browser Helper Object

A BHO was developed to extend the web browsing capabilities of clients using Microsoft Internet Explorer (IE). A BHO is an in-process Component Object Model (COM) component that IE will load on startup, can detect browser events, access and modify the browser toolbar and menus, create new windows, and install hooks to monitor browser actions (Esposito 1999). User settings are stored in a text file on the client mobile device, and can be altered at any time by the user. All document requests sent by the client browser are intercepted by the BHO and modified before being sent to the request handler. Settings are appended on the first request, and any subsequent request made after the user preferences have been altered. A unique identifier is used to inform the proxy that settings have been appended to a request. The BHO implemented listens for the *`BeforeNavigate2()`*[xx] event, which is fired when IE requests a new URL. The navigation is then cancelled, and a new request is created by appending the user preferences to the original request. This new URL request is then sent using the *`Navigate2()`*[xxi] method, which is the standard IE method to transmit web requests. Figure 4-6a shows an example web request. Figure 4-6b shows the request with a unique identifier and user settings appended, in this case a '#' symbol has been used as the identifier. User preferences can take any

[xx] http://msdn.microsoft.com/workshop/browser/webbrowser/reference/events/beforenavigate2.asp
[xxi] http://msdn.microsoft.com/workshop/browser/webbrowser/reference/methods/navigate2.asp

form, and correspond to any adaptation modules required. However, the proxy server must be capable of identifying and utilizing the attached preferences before fetching the required content document. This is the only section of the system that requires installation of specialized client-side software. The intelligence could be placed on the server-side if required by developing a mobile web application. In this way users could utilize a web site to control their personalized settings. It is not a necessity to have the BHO installed on a client device in order for the client to use the adaptation system.



http://www.ru.ac.za/

http://www.ru.ac.za/#(usersettings)

(a) **(b)**

Figure 4-6. (a) An example web request. (b) The web request with user preferences attached.

4.2.2 SQL Database and Client Settings

The proxy server maintains a SQL database, containing session information about each user's adaptation preferences. The server side personalization module was developed using the J2SE. MySQL[xxii] 4.1 was the database engine implemented, and the JDBC API 3.0 was used to handle the connection to the database. On receiving a request the proxy server analyses the URL requested, searching for an identifier indicating that user preferences have been appended. Preferences are removed from the request, which is then handled in the normal way. The settings are stored in the database, and the IP address of the client device is used as the primary key. The IP address is obtained using the `java.net.ServerSocket.getInetAddress()` method[xxiii]. This method returns the local IP address of the machine connected to the receiving server socket. When client devices send requests without settings appended, the preferences are obtained from the database using the client IP address. If the proxy server does not discover any adaptation settings appended to a request or match the client to any devices stored in the database, the content returned to the client is adapted according to a default set of adaptation options maintained on the server.

[xxii] http://www.mysql.com/

[xxiii] http://java.sun.com/j2se/1.4.2/docs/api/java/net/ServerSocket.html#getInetAddress()

4.3. Responsive Content Adaptation

Previous work on content adaptation has had an influence on network performance metrics (Lyijynen *et al.* 2003). This is due to the fact that during the extraction process data are discarded or manipulated, usually resulting in a reduction in size of the content page being processed. However, this influence has been a side effect and not measured or controlled. The extent to which content adaptation influences performance metrics, such as bandwidth usage, is evaluated. The extraction framework is extended to allow for adaptation to network conditions as a method for increasing the available bandwidth in legacy networks.

The aim of this module is to investigate the feasibility of implementing an automated content adaptation system, where adaptation settings are altered according to changes in network conditions. The approach will be evaluated on the ability to improve the QoS in a wireless network. This should be done by reducing congestion on the network, and thereby reducing the packet loss on the network. By reducing the number of packets, bandwidth usage should be reduced. Unlike traditional QoS techniques, this method functions at the data layer. Performing content adaptation will cause a delay due to processing costs on the server. The system requires that the benefits created, such as reduced bandwidth usage, outweigh the additional server side processing costs incurred. By reducing the amount of data delivered to a client in a resource sharing wireless network, it would be possible to provide each client with additional bandwidth thereby improving the QoS, or allow additional clients to enter the network and maintain a similar QoS for the existing clients.

Figure 4-7 illustrates an overview of the responsive adaptation system architecture. The network monitor maintains various network statistics, which are made available to the adaptation modules. The proxy server forwards requests to the adaptation modules, where settings are applied according to the current network conditions.

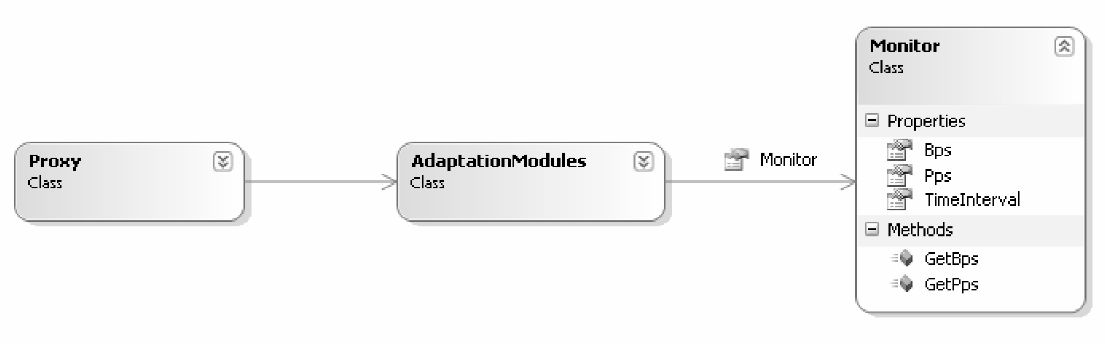


Figure 4-7. System architecture of the responsive content adaptation approach.

4.3.1 Monitoring Network Conditions

The goal of the network management extension is to maintain a specified level of network usage. For this reason, an essential part of the system is the ability to monitor network conditions, so that the desired level of usage can be maintained. In particular, the focus is on the amount of traffic on the network (in bytes). WinPcap[xxiv], an open source packet capture and network analysis library for Windows, was utilized to achieve this (Degioanni 2000). Since the WinPcap library was developed for use with C++, it was necessary to create a Java wrapper class in order to expose the methods required by the network monitor. The monitor class contains values for the average bytes transmitted per second, the average packets transmitted per second, and the number of connections made in a specified time period. WinPcap provides access to statistics for a specific network interface card (NIC). The `pcap_open()`[xxv] method is used to initiate a NIC, and the `pcap_setmode()`[xxvi] method is used to set the interface to statistics mode. `pcap_open()` opens a generic network source, and returns a pointer to the source. `pcap_setmode()` sets the working mode of the NIC, `MODE_STAT`[xxvii] sets the mode to statistical which allows the TCP load of the device to be monitored. The number of bytes sent and received are measured and continuously added together. Once the specified time period has elapsed, this running total is divided by the time, providing the number of bytes per second (bps), and is then reset and the process restarted. The same is done for the number of packets sent and received, and the number of connections made.

[xxiv] http://www.winpcap.org/
[xxv] http://www.winpcap.org/docs/man/html/group__remote__func.html#a3
[xxvi] http://www.winpcap.org/docs/docs31/html/group__wpcapfunc.html#ga48
[xxvii] http://www.winpcap.org/docs/docs31/html/group__wpcap__tut9.html

4.3.2 Adaptive Logic

The responsive adaptive algorithm monitors network conditions and automatically applies adaptation techniques at set threshold values, and the goal is to maintain a maximum level of network bandwidth usage, namely the threshold value. The network monitoring module can be used as an extension to any adaptation module, providing current network usage statistics, which can then be used to adjust adaptation settings. Removing rich content should result in a drastic drop in bandwidth usage. Abdelzaher and Bhatti (1999) propose a solution to solve the over and underutilisation problem, where content is removed from every n^{th} document (Chapter 2, Section 2.3.2).

The adaptive logic functions as follows:

- The network administrator specifies the desired maximum and minimum levels of network usage (in bytes), and the initial fraction of content documents to transform, which is the value *n* that defines the n^{th} document to be transformed. Initially, no content documents are transformed.
- The network monitor records network usage levels, and informs the content adaptation modules when the desired maximum level is reached.
- Every *nth* document is transformed by an adaptation module, reducing the load on the network. If network usage continues to increase, the value of *n* is reduced until *n*=1, where every document is transformed. If network usage begins to decrease at any stage, and falls below the desired minimum usage level, the value of *n* is increased until the initial specified value is reached. If network usage still continues to fall, *n* returns to 0, where no documents are modified.

4.4. Summary

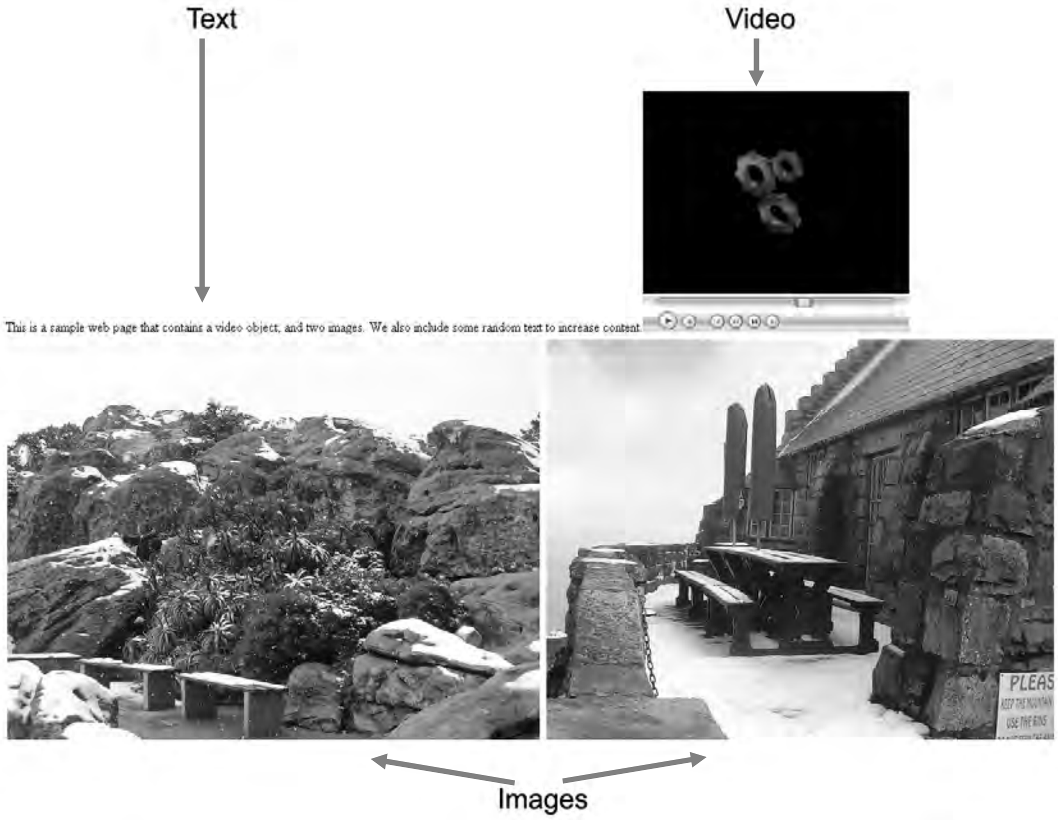
This Chapter described in detail the implementation of three content adaptation techniques, tag extraction, text summarization and spatial representation. An approach to personalization of extraction settings was detailed and a system for implementing responsive adaptation was described. Tag extraction can be utilised to remove, replace, or modify content within HTML and XML documents. The goal of tag extraction is to reduce document size and dimension. Text summarisation divides content pages into sections, and provides a tree view which can be used to navigate the documents. The approach aims to reduce document dimension. Spatial representation also identifies content document segments, but provides a graphical representation of the layout as a means of navigation. The personalisation module aims to provide users with the ability to specify adaptation settings, allowing them to customise the appearance of the content documents. A network management module has been developed that monitors network conditions, and attempts to maintain a specified level of network usage. These framework modules and extensions will be used to extend existing electronic display systems according to the framework proposed in the previous chapter, which will then be used to evaluate the framework. The next Chapter evaluates the modules and extensions implemented according to effectiveness and performance.

Chapter 5

MODULE EVALUATION

Chapter 4 described the implementation of three content adaptation modules (tag extraction, text summarisation, and spatial representation) and two adaptation extensions (personalisation and responsive adaptation). This Chapter focuses on evaluating the individual modules in terms of performance and effectiveness, while Chapter 6 evaluates extending electronic display systems, by making use of the modules and extensions, according to the adaptation framework. Effectiveness of a module is deemed to be the correctness of a document produced as output of an adaptation module, and is not a measure of usability. Thus effectiveness is evaluated by considering the intended outcomes of a module in terms of processing overhead and correct representation of a document. The responsive adaptation framework (Chapter 3, Section 3.2) defines a set of requirements. One of these requirements is a low response time overhead for the adaptation modules and the extensions implemented for use with the framework. Content adaptation modules and framework extensions provide benefits, such as reduced document dimensions and network management. However, these benefits would be negated if the processing cost of the modules and extensions significantly increased the round trip time of a client request, that is, the time taken for a request issued by a client to be fulfilled. Therefore, the main aim of this Chapter is to evaluate the performance, in terms of response time overhead, of the adaptation modules and framework extensions in order to determine whether the implementations incur a minimal processing cost overhead on the total request time of a document, or if the processing overhead negates the potential benefits.

In order to evaluate the performance of the implemented modules and extensions, a sample HTML document was created that can be used by each adaptation module (Figure 5-1). The document consists of text (125 characters), an embedded video object (200kb), and two images (50kb each). This range of content provides a means to test the effects of the adaptation modules, which can be used to extract certain content types, on the processing overhead of the approaches and the influence on network metrics. All experiments were carried out using an Intel Pentium 4 3.00GHz processor, with 1.00GB of RAM, 100Mbps network interface card, running Microsoft Windows XP Professional Version 2002, with Service Pack 2, as the operating system. In order to establish a base request round trip time a C# application was developed that requested the sample document and recorded the time taken to the first byte. These times were then used as a control to evaluate the processing overhead cost caused by each of the adaptation modules. Crunch 2.0 was used as the proxy server, and 10 requests were sent to obtain an average value. In order to determine the effect of increasing document size on processing cost of the adaptation modules, the tags in the sample document (control) were replicated by powers of 2 resulting in documents with 4, 8, 16, 32, 64 and 128 tags respectively. In terms of content on each page, at 4 tags the document contained 1 text string, 2 images, and one video object, while at 128 tags the document contained 128 text strings, 256 images, and 128 video objects. Again 10 requests were made for each document size. Figure 5-2 shows the results of sample document in which no adaptation modules were applied, and indicates a linear relationship ($r^2 = 0.99$, $y = 11.72x$) between the number of tags and the time taken to process a document. This increased from an average of 55.02ms for a 4 tag document, to 1501.77ms for a 128 tag document. These times represent a worst case scenario, as the testing was carried out on the machine local host, and not on the comparatively slower wireless networks.



This is a sample web page that contains a video object, and two images. We also include some random text to increase content.

Figure 5-1. The sample HTML content document created for test purposes.

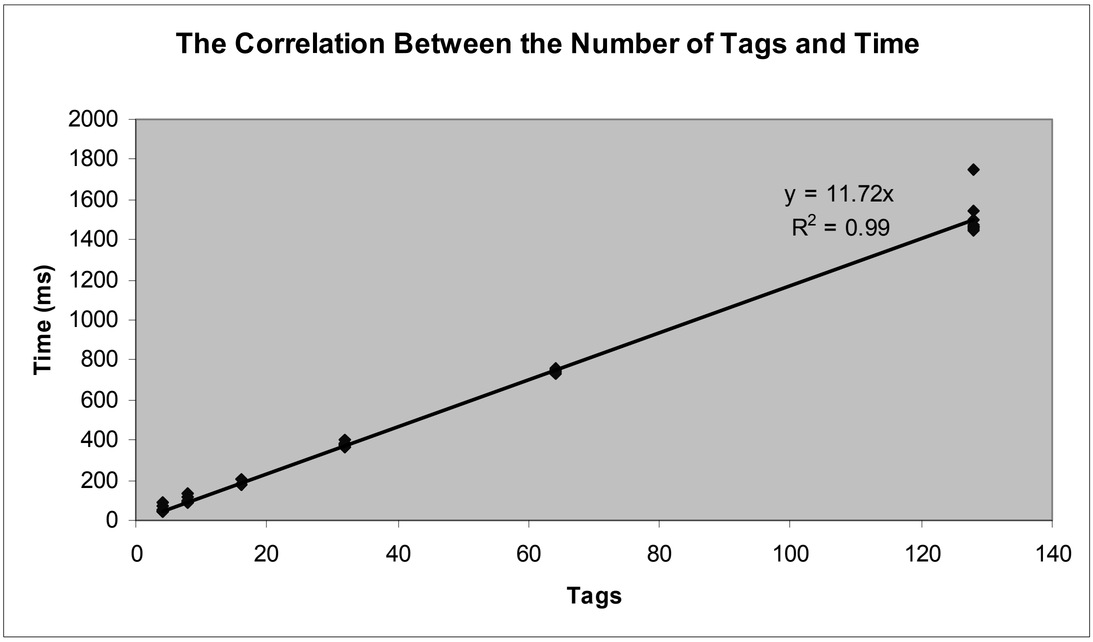


Figure 5-2. The correlation between the number of tags and time taken to process the sample document (control) without any adaptation modules.

5.1. Content Adaptation Modules

Each of the three content adaptation modules were tested using the same sample HTML document created previously (Figure 5-1) and again replicated by powers of 2 resulting in documents with 4, 8, 16, 32, 64 and 128 tags respectively. The goal was to establish the performance overhead associated with each module, and to compare this overhead to the average time to first byte with no module in use. A test C# application was used to create a virtual client, which sends requests for the sample document once every 5 seconds. The time taken by the modules to process the requested document was recorded.

5.1.1 Tag Extraction

The aim of tag extraction is to remove, replace, or modify specified content tags within an HTML or XML document, thereby reducing the dimensions of the document and reducing the number of bytes sent across the network. In order to identify tags, the entire document needs to be parsed. It is obvious that the effectiveness of the module in reducing the size, in bytes, of a document, is determined by the content makeup of the document. That is, the approach would be successful for documents containing rich multimedia content, such as images and video, since this content would constitute a large portion of the document size and would greatly reduce the size when removed or modified. Alternatively, the approach would be relatively ineffective for documents containing little or no rich content, as the entire document would still be analyzed with little or no benefit.

5.1.1.1 Performance

Since the entire document is parsed, an important aspect of the module is the processing speed required, as this would increase the round trip time of a request and should therefore be minimal. In order to evaluate the speed of the tag extraction module, timing mechanisms were installed to measure the length of time required to process a document. Timing was initiated before the function call to the tag extraction module, and stopped when a response was received. Since the tag extraction module was implemented using Java, the `System.nanoTime()`[xxviii] method was used to gain a starting and ending time value. `System.nanoTime()` returns the value of the system timer in nanoseconds.

Figure 5-3 illustrates the results from a regression analysis of the data obtained by measuring the time taken for the tag extraction module to process a document with continually increasing numbers of tags. There was a linear relationship ($r^2 = 0.99$, $y = 11.84x$) between the number of HTML tags and the time taken to process a document. The request time increased from 55.48ms for a 4 tag document, to 1519.51ms for a 128 tag document, with an average increase of 0.72% from the control times recorded (Figure 5-2).

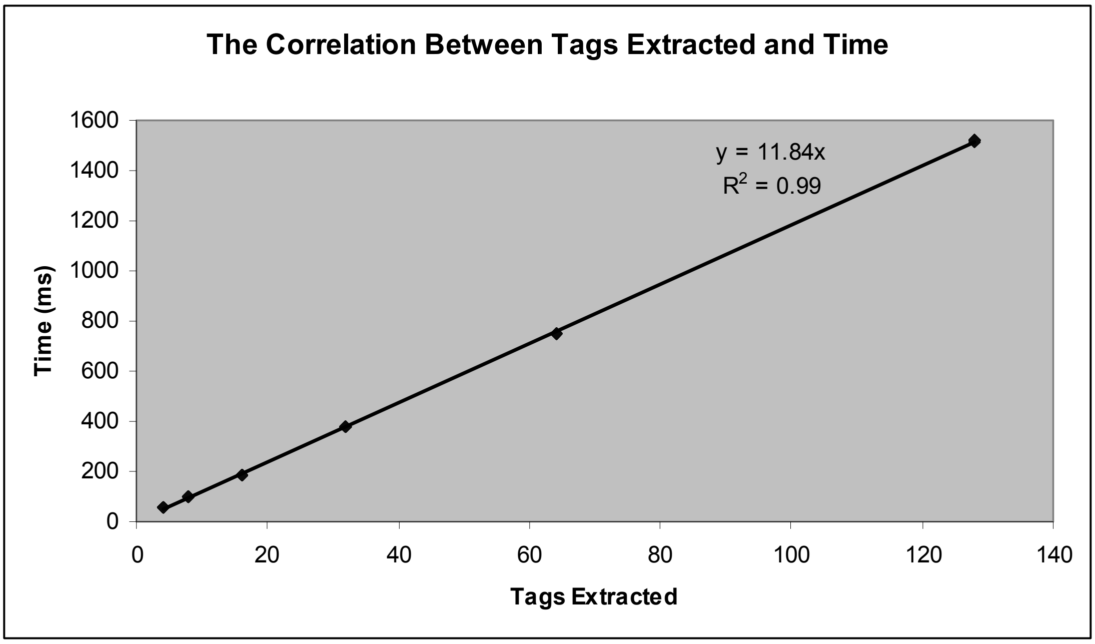


Figure 5-3. The correlation between the number of tags extracted and time taken.

Figure 5-4 illustrates the relationship between the number of different types of tags to be extracted, and the time taken for the extraction module to process a document. The number of tag types to be extracted is initially 1, and increases in powers of 2, until 8 tag types are extracted. There was a linear relationship ($r^2 = 0.99$, $y = 11.79x$) between the number of HTML tag types and the time taken to process a document. This increased from an average of 55.19ms for 1 tag type, to 377.63ms for 8 tag types.

[xxviii] http://java.sun.com/j2se/1.5.0/docs/api/java/lang/System.html#nanoTime()

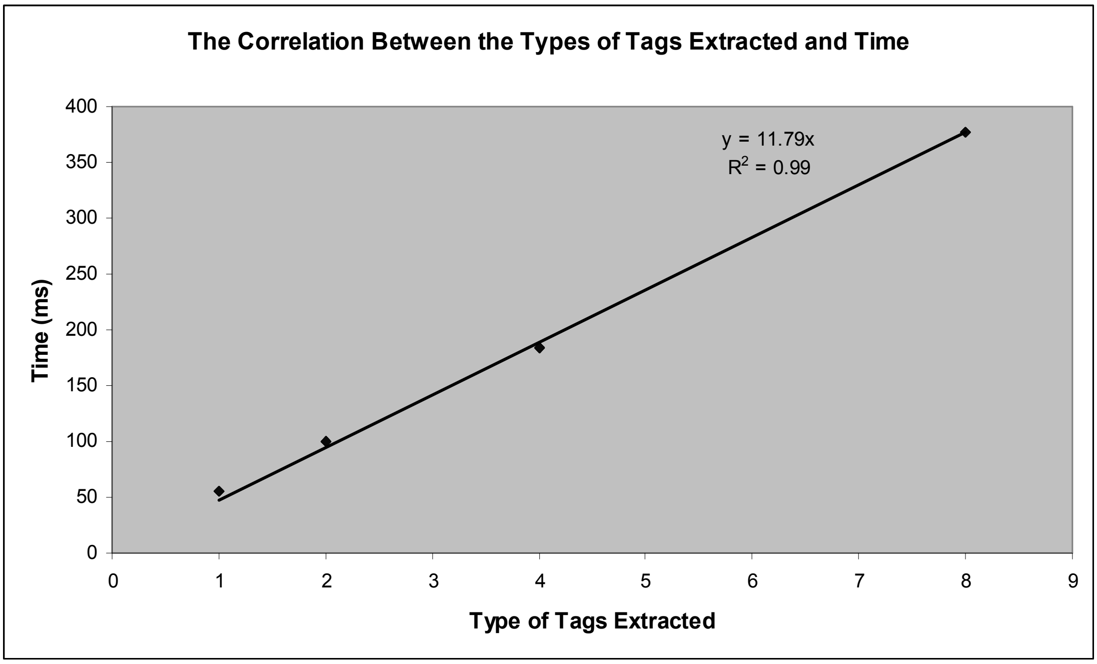


Figure 5-4. Time taken as a function of the number of different types of tags extracted.

The time taken for the tag extraction module to manipulate a document corresponds linearly to the number of tags in the document, and to the number of tag types which require extraction. This approach is not limited by the HTML/XML structure of a document, since the algorithm compares each tag to a list of tags to be extracted. Timing of the approach indicates that computation times are negligible, adding a 0.72% processing overhead to the round trip time of the sample document. In addition, an increase in processing time was observed with increasing tag numbers and types.

5.1.1.2 Effectiveness

The tag extraction approach makes use of a parser to traverse the HTML/XML tree and to identify relevant tags that require some form of manipulation. The parser traverses every tag in the document, and the tag extraction implementation functions correctly if the HTML or XML document is well formed. This particular implementation of the tag extraction approach replaces specified tags with links to the content, which may remove the intended context of the content as it is viewed away from the surrounding content contained within the document. Figure 5-5 illustrates the result of applying the tag extraction algorithm to the example HTML document (Figure 5-1), in this case all images were replaced by links. Although it is not possible to test for all possible document structures, it can be assumed that the tag extraction algorithm functions correctly as the algorithm treats a document as a tree of tokens, which is traversed and every token is compared to a specified string. This full tree traversal and simple string comparison of tokens ensures that correctly spelt tags will not be overlooked. In order to show this an experiment was conducted where the order of certain tags in the sample HTML document was altered. The sample document has the form <HTML><BODY>…</HTML></BODY>, and was altered to <BODY><HTML>…</BODY></HTML>. Although this new sample document is structurally incorrect, the tag extraction module still produced the same document as the original experiment when extracting images (Figure 5-5). However, errors occur when the order of the tag to be extracted is incorrect, or the tag is incorrectly spelt in the document. For example, the text tag to be extracted should be in the form <P>…</P>, but is actually in the form </P>…<P>. In this instance the tag will not be successfully extracted as there is no <P> node in the DOM tree, but instead there is a </P> node, which will not match the specified search string of "<P>". Similarly, when extracting an image object the tag should read <IMG…>, but if spelt incorrectly (eg. <IMF>) the tag extraction algorithm will not identify the tag as an image object. These issues are considered trivial since the original document will not display correctly with the incorrect spelling of tag names, and is thus not an error of the extraction module. A possible improvement to the algorithm is the automatic identification and correction of document structure and syntax.

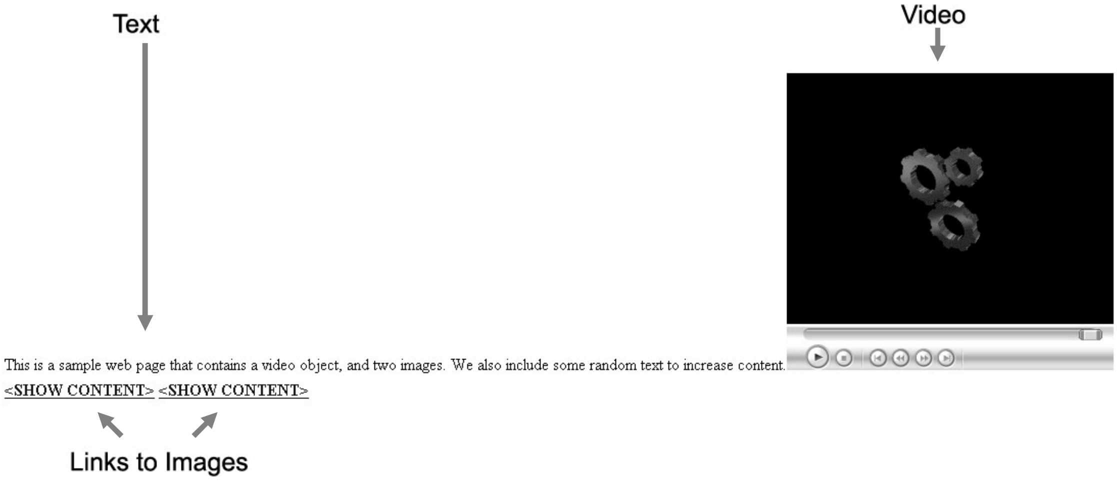

This is a sample web page that contains a video object, and two images. We also include some random text to increase content.

Figure 5-5. Results of the tag extraction algorithm on the sample HTML document (Figure 5-1).

Experiments conducted using the tag extraction algorithm showed the approach to be successful for all possible correct HTML and XML structures. However, with this implementation it is not possible to extract within embedded content such as Macromedia Flash, but only to either extract all or none of this content. A second limitation is introduced by the use of XML, where it is possible to name tags using unique, non-standard identifiers. This is not an issue with HTML documents since tag names are standardized. This approach is very effective because all tags are parsed, and the tag names matched against a list, guaranteeing successful processing of well formed documents. Although in this example tags were merely replaced by links, it would be possible to apply more advanced techniques, such as reducing image and video quality.

5.1.2 Text Summarization

Text summarisation provides a mechanism to navigate a document through the use of a tree view structure. This is more appropriate for small screen devices, as the documents are divided into smaller viewable segments, and the tree view object provides a convenient method of navigating a document. The approach has the potential to reduce the amount of data on the network if a user does not access all content segments displayed on the tree view.

5.1.2.1 Performance

Since the entire document is parsed, an important aspect of the module is the processing time required, as this would increase the round trip time of a request and should therefore be minimal. In order to evaluate the speed of the text summarization module, timing mechanisms were installed to measure the length of time required to process a document. Timing was initiated before the function call to the module, and stopped when a response was received. The module was developed using C#, and thus the C# DateTime[xxix] structure was used to measure a start and end time.

[xxix] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html /frlrfsystemdatetimeclasstopic.asp

`DateTime.Now()` returns the current system time, and can be expressed in Ticks, which are defined as 100 nanoseconds.

Figure 5-6 shows the timing results of generating tree view controls for a document with an increasing number of page segments. There was a linear relationship ($r^2 = 0.99$, $y = 12.02x$) between the number of page segments and the time taken to process a document. The request time increased from 60.27ms for a 4 tag document, to 1537.62ms for a 128 tag document, with an average increase of 4.98% from the control times recorded (Figure 5-2).

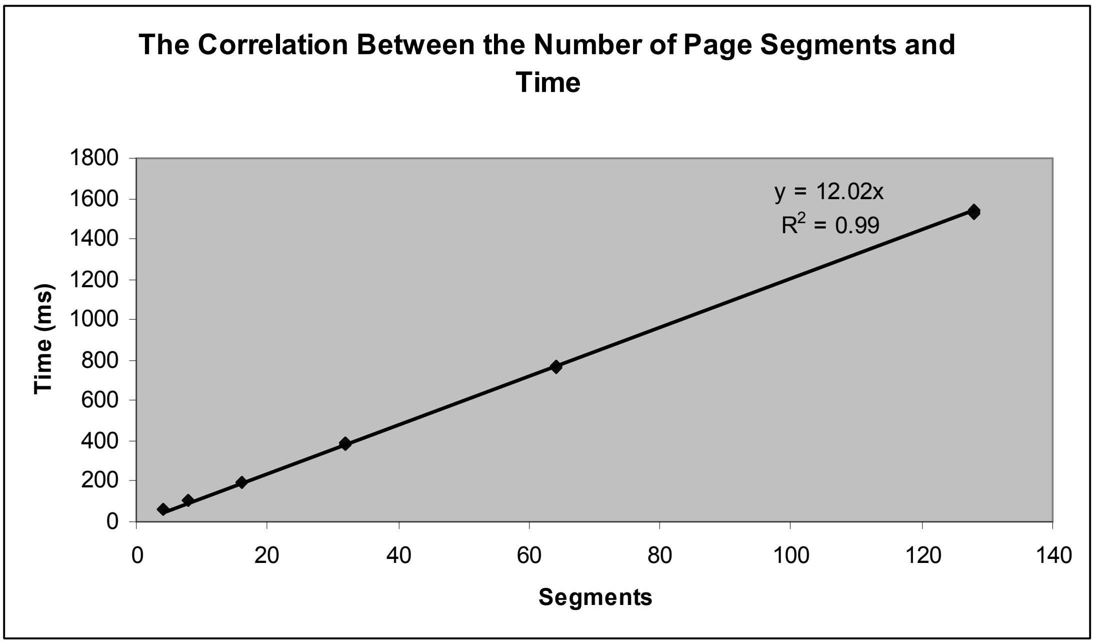


Figure 5-6. Results of generating a tree view structure of different length.

Timing of the approach indicates that computation times are low, adding a 4.98% processing overhead to the round trip time of the sample document. Again, an increase in processing time was observed with increasing number of segments in a document when generating a tree view control.

5.1.2.2 Effectiveness

In order to create a meaningful tree view representation of a content document, it is necessary to either correctly infer the intended layout, or to design documents with a certain structure (Chapter 4, Section 4.1.2). Figure 5-7 shows a tree view control generated from the sample HTML document (Figure 5-1). The tree view provides a list of the document segments, and selecting one of these links will display the relevant content. In the example an image is selected (Figure 5-7). Text summarisation functions by locating HTML or XML tags within a document that have been designated as identifying document segments. For this reason, the approach is limited to a pre-defined subset of documents, which adhere to specified design constraints, such as conforming to naming conventions for tags. Large documents, broken into many segments, will require scrolling of the tree view control in order to access all the content contained within the document.

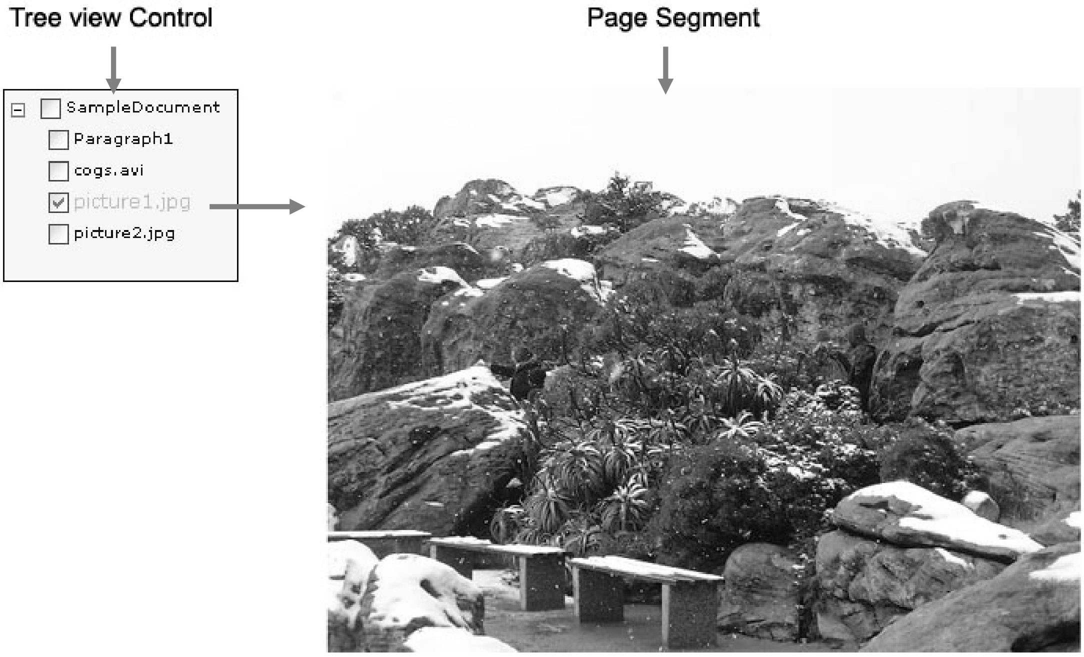


Figure 5-7. A tree view navigation structure of the sample HTML document (Figure 5-1) created by the text summarization module.

The text summarisation module is restricted to a subset of HTML or XML documents that adhere to tag naming conventions in order to correctly identify document segments. This text summarization implementation does not attempt to infer document structure, but assumes a design pattern has been followed and makes use of specified tag names to identify document segments.

5.1.3 Spatial Representation

Spatial representation is similar to the text summarization approach in that content documents are divided into segments, allowing the content to be navigated and displayed in a convenient way on small screen devices. However, spatial representation provides a graphic representation of the document and the objects contained therein, instead of the tree view control used in text summarization. The image produced during the spatial representation process can be scaled to reduce size without losing substantial quality, since it is a simple image. The purpose of spatial representation is to provide a convenient method of document navigation and to remove the need to download an entire document when a page is accessed.

5.1.3.1 Performance

As with the tag extraction module, the entire document is parsed in order to identify specific tags. This would increase the round trip time of a request, and thus the processing cost of the module should be minimized. The spatial representation module was implemented using Java, therefore the *`System.nanoTime()`* method was used again in order to evaluate the processing overhead caused by the method. Timing was initiated before the function call to the spatial representation module, and stopped when the process was complete.

Figure 5-8 shows the timing results of generating spatial representations of a document with an ever increasing number of content blocks. There was a linear relationship ($r^2 = 0.99$, $y = 13.48x$) between the number of content blocks and the time taken to process a document. The request time increased from 68.70ms for a 4 tag document, to 1707.45ms for a 128 tag document, with an average increase of 20.42% from the control times recorded (Figure 5-2).

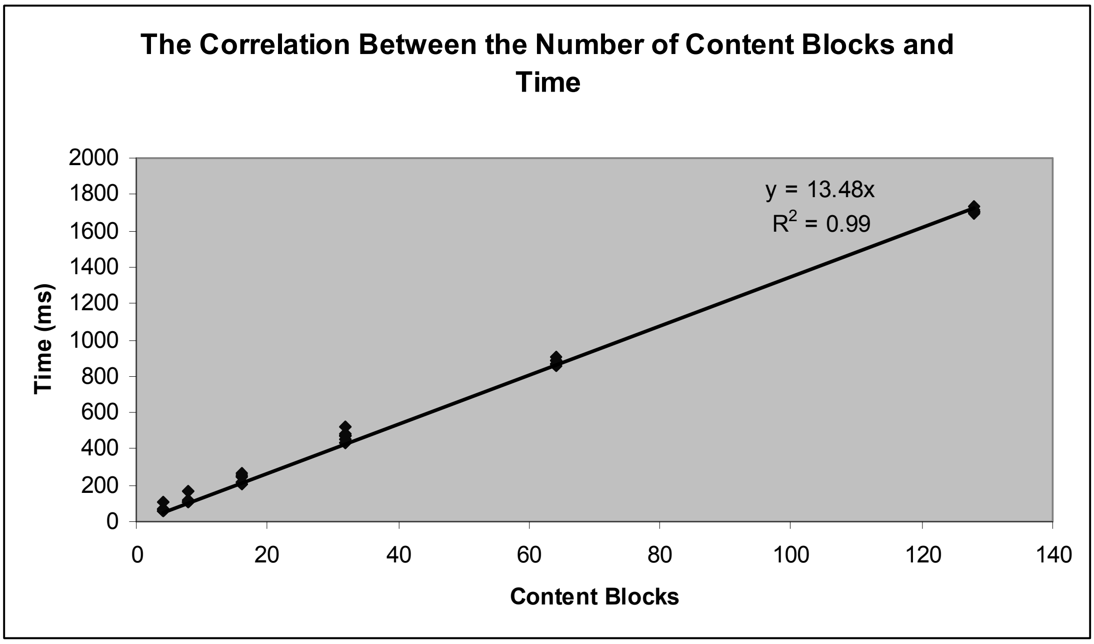


Figure 5-8. Time as a function of the number of blocks identified.

Timing of the approach indicates that computation times are fairly costly, adding 20.42% processing overhead to the round trip time of the sample document. Again, the time taken for the module to transform a document shows a linear correspondence to the number of content blocks in the document.

5.1.3.2 Effectiveness

The spatial representation extension parses HTML/XML documents and identifies content blocks by specified tag names. This approach has no method of inferring document layout, and thus documents must be structured using the appropriate tag names. Figure 5-9 illustrates the effects of transforming the sample HTML document (Figure 5-1) using the spatial representation algorithm. The blocks shown in Figure 5-9 correspond to the content displayed in Figure 5-1. A HTML image map is created by the module which defines the boundary coordinates for each content block in the image. Thus when a client selects a content block from the representation, the browser navigates to the corresponding content defined by the image map.

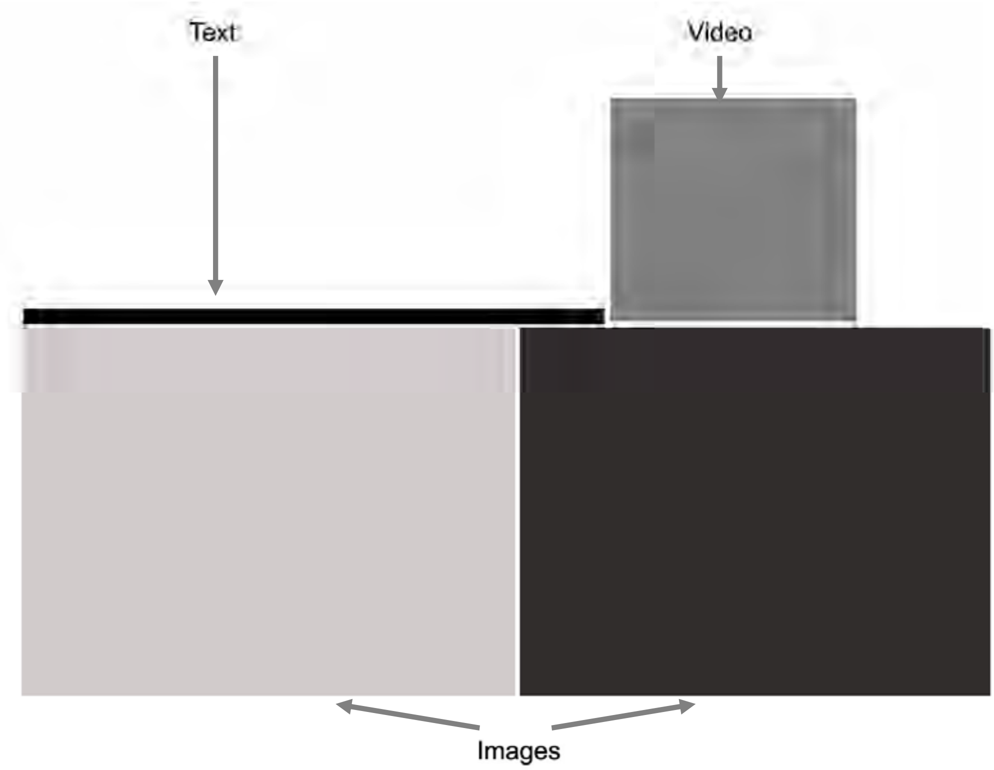


Figure 5-9. A spatial representation of the sample HTML document (Figure 5-1).

Spatial representation provides a convenient means to navigate a content document, but is restricted to a subset of HTML or XML documents. This adaptation module does not automatically detect document structure, but assumes a design pattern has been followed and makes use of specified tag names to identify content blocks.

5.2. Personalisation

The personalization extension provides users with the ability to specify adaptation preferences, removing the automatic adaptation functionality from the system. Personalization provides users with a method of specifying settings for adaptation modules, such as tag extraction, extended the functionality of these modules. This approach would allow users to customize documents, as each client would receive content formatted according to their settings. Preferences are stored locally on the client device and are sent to the proxy server with the first request, and any subsequent requests where the settings have been modified. Settings are stored in a MySql database on the server side, and are retrieved by the system with inline SQL statements. A Browser Helper Object (BHO) is installed on the client device as an extension to Internet Explorer (IE). The BHO intercepts web requests made by the client, and appends the client preferences to the request. The server then removes these settings from the web request, and stores the preferences in the database. Client devices without the BHO installed can still access content as default adaptation settings are employed if no settings are appended to the request or stored in the database.

5.2.1 Performance

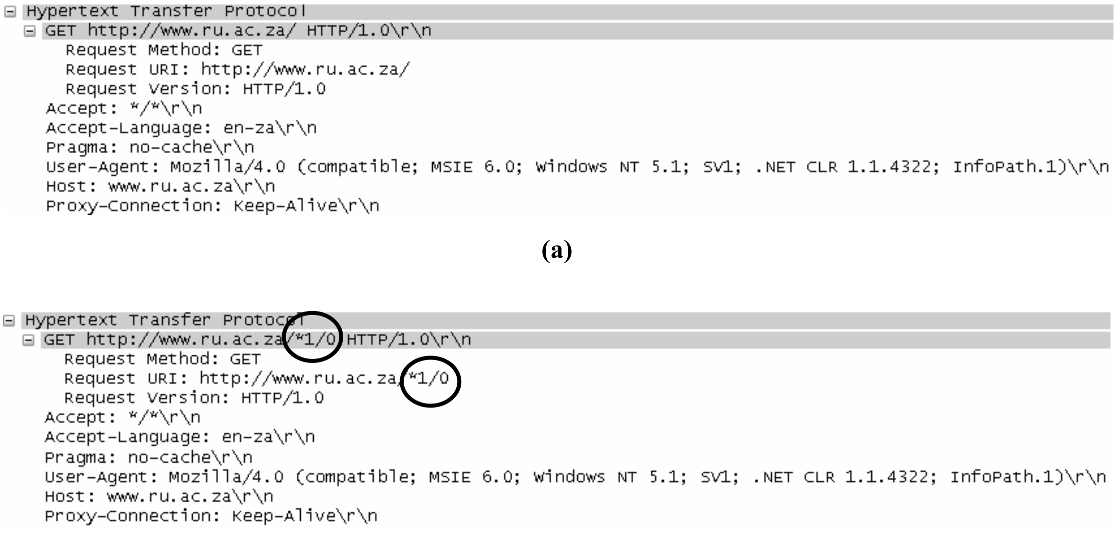
The only cost to the round trip time of the request is the time taken to extract the settings from a user request, store these settings in the database, and then to retrieve the settings from the database. The sample HMTL document (Figure 5-1) was requested and user preferences were requested. Request times were measured using the *`System.nanoTime()`* method, providing a means to evaluate the processing overhead caused by the method. Start and end times were measured for extracting user settings from a web request, storing these settings, and retrieving the settings. These times were then added together to measure the total added cost to the user request. Timing of the approach indicates that computation times are negligible, adding an average processing overhead of approximately 5.96% to the round trip time of the sample document. This overhead equates to an additional 3.28ms request time, and includes extracting the settings from the request, storing the settings in the database, and retrieving the relevant preferences for content adaptation.

5.2.2 Effectiveness

The personalisation approach appends user-specified adaptation settings to a web request sent to the proxy server. Ethereal Network Protocol Analyser[xxx] was used to capture packets sent to the proxy server. Figure 5-10a shows part of an HTTP GET request sent to the server, and Figure 5-10b shows the same request sent with the personalisation module in place. The personalisation module was used to append adaptation settings defined by the user (Figure 5-10b). In this example the appended settings (*1/0) are used for tag extraction, where the * is used to identify the settings from the URL, the 1 means that images should be displayed, and the 0 means that

[xxx] http://www.ethereal.com

video should be extracted. The captured packets show that the personalisation module successfully appends user defined settings to requests sent to the proxy server. Settings are configured on the client-side, and can be adjusted to suit various adaptation modules. The settings are extracted on the server-side, and used as input for the adaptation modules when adapting documents requested by the relevant user.



```
⊟ Hypertext Transfer Protocol
  ⊟ GET http://www.ru.ac.za/ HTTP/1.0\r\n
      Request Method: GET
      Request URI: http://www.ru.ac.za/
      Request Version: HTTP/1.0
    Accept: */*\r\n
    Accept-Language: en-za\r\n
    Pragma: no-cache\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; InfoPath.1)\r\n
    Host: www.ru.ac.za\r\n
    Proxy-Connection: Keep-Alive\r\n
```

(a)

```
⊟ Hypertext Transfer Protocol
  ⊟ GET http://www.ru.ac.za/*1/0 HTTP/1.0\r\n
      Request Method: GET
      Request URI: http://www.ru.ac.za/*1/0
      Request Version: HTTP/1.0
    Accept: */*\r\n
    Accept-Language: en-za\r\n
    Pragma: no-cache\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; InfoPath.1)\r\n
    Host: www.ru.ac.za\r\n
    Proxy-Connection: Keep-Alive\r\n
```

(b)

Figure 5-10. (a) A HTTP GET request. (b) The same request with adaptation settings appended using the personalisation approach. Appended settings have been circled for clarity.

The personalisation extension was implemented successfully for devices running IE, and the possibility exists to extend the system to function for different browser software. This approach provides users with the ability to specify unique adaptation settings, thus providing an improved user experience. Timings indicate that the processing overhead of the personalisation extension is negligible, increasing the request time of the sample HTML document (Figure 5-1) by 5.96%.

5.3. Responsive Adaptation

The responsive adaptation extension aims to modify adaptation settings in order to influence network conditions, reducing the bandwidth usage in the network. A network monitor measures the levels of network usage, and this information is then used to adjust the content adaptation modules to reduce strain on the network. The responsive adaptation extension alters tag extraction settings in order to reduce network traffic (Chapter 4, Section 4.3). This is due to the fact that during the tag extraction process content is removed from the document, usually resulting in a reduction in size, in bytes. Testing was carried out to evaluate the effectiveness of the responsive extension in terms of maintaining a set threshold of network usage.

5.3.1 Performance

The performance cost to the round trip time of requests is the time taken to retrieve the network statistics from the network monitor, and the adaptive logic function which utilizes these statistics to control the tag extraction module. Once again, the *`System.nanoTime()`* method was implemented in order to measure the processing overhead caused by the method. The sample HMTL document (Figure 5-1) was requested, and the time taken to perform the network management tasks was measured. Timing of the approach indicates that computation times are negligible, adding an average processing overhead of approximately 0.16% to the round trip time of the sample document. This overhead equates to an additional 0.09ms request time.

5.3.2 Effectiveness

A sample HTML document was created in order to test the responsive adaptation module (Figure 5-1). Load testing was initially carried out using Microsoft Application Centre Test (ACT), and ANTS Load[xxxi] from Red-gate Software[xxxii]. These applications were not effective as IE is used as the browser to request pages, and the content was cached, removing the effectiveness of the tag extraction module. In order to solve this issue a test application was developed using C# that creates multiple threads, where each thread requests a specified HTML document every 5 seconds. In this case, the document requested was the sample HTML document (Figure 5-1) created for test purposes. The document is retrieved by using the *`WebRequest`*[xxxiii] class, which allows an application to request content from a network in a protocol-agnostic manner. This retrieved document is then parsed and content objects identified and retrieved. Threads can be added or removed at will.

[xxxi] http://www.red-gate.com/products/ants_load/

[xxxii] http://www.red-gate.com/

[xxxiii] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html /frlrfsystemnetwebrequestclasstopic.asp

Implementing a custom load testing application removed the caching issue, allowed the frequency of requests to be specified, and provided a platform where the number of simultaneous clients could be incremented and decremented as required.

Figure 5-11 shows the results of implementing the responsive adaptation extension, in this case an ever increasing number of clients access the test document. Adaptive logic of the responsive approach was implemented as defined in Chapter 4, Section 4.3.2. The variable *n*, that specifies which initial fraction of documents to adapt, was set to 5. The upper threshold level was set to 1000000 bytes, and the lower threshold to 800000 bytes. After 3 users have accessed the system the threshold is reached and the system responds by instructing the tag extraction module to remove all video content for every document (Point *a*, Figure 5-11). This causes a drastic decrease in bandwidth usage (74.31%) at point *b* in Figure 5-11, resulting in underutilisation of the network resources owing to unused bandwidth (Chapter 2, Section 2.3.2), which slowly increases again as more users are added.

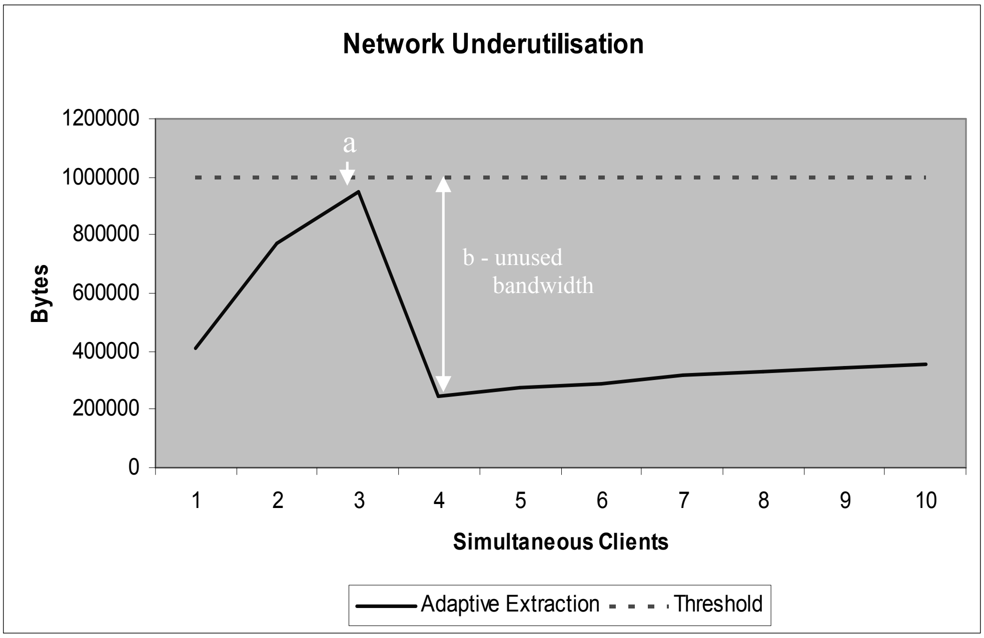


Figure 5-11. Responsive content adaptation showing the underutilization of network resources once tag extraction has been implemented, removing the video object from the sample document (Figure 5-1).

Figure 5-12 shows the results of varying the levels of *n*, where *n* is the n^{th} request received by the system, in order to maximise the levels of network usage while still adhering to the threshold specified. The threshold level is reached again after 3 clients, but in this case tag extraction is only applied to every 5th document requested (Point *a*, Figure 5-12), resulting in a less dramatic reduction in network usage. Thus the amount of unused bandwidth is less than as shown in Figure 5-11. As network levels increase further the value of *n* is reduced, first to 3 (Point *b*, Figure 5-12), and finally to 2 (Point *c*, Figure 5-12) according to the adaptive logic (Chapter 4, Section 4.3.2). This approach is still successful in maintaining network usage below the threshold, but alleviates the underutilisation issue.

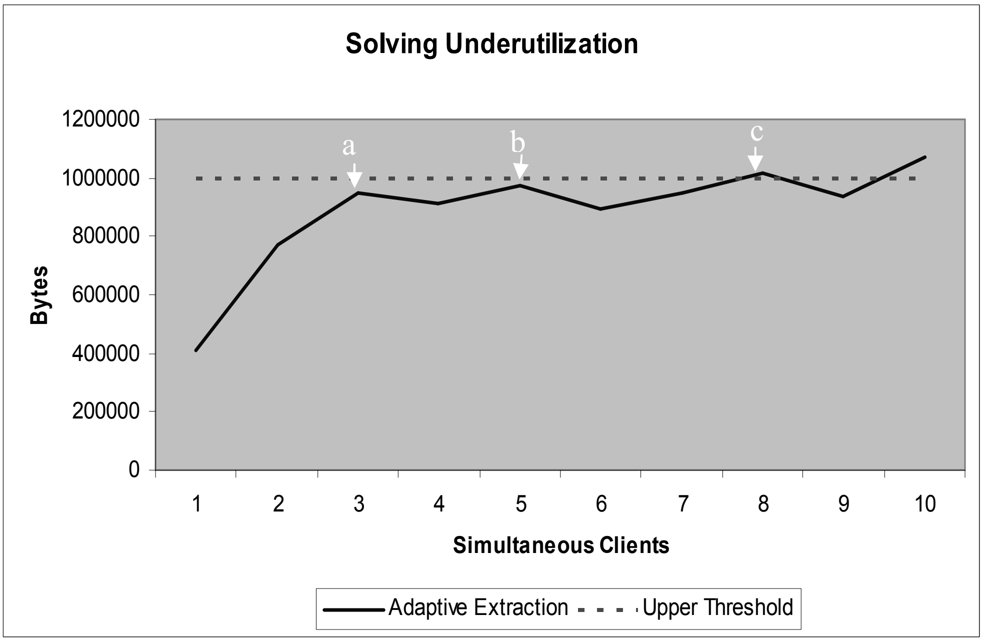


Figure 5-12. Responsive content adaptation for solving the underutilization problem. Tag extraction is applied to every nth document in order to maintain the desired level of network usage. The value of n is reduced as network usage increases.

Figure 5-13 illustrates how the system responds to decreasing levels of network usage. In this case a lower threshold of 800000 bytes was set. As the number of clients on the network decreases, so does the bandwidth usage. In response, the system increases the value of *n* when the lower threshold is reached. Initially *n*=2, and is first increased to 3 (Point *a*, Figure 5-13), then to 5 (Point *b*, Figure 5-13), and is finally made 0 (Point *c*, Figure 5-13) so that no documents are manipulated. At the stage where 2 users are active, network usage falls below the lower threshold even though *n*=0. Performance evaluation showed an average processing overhead of 0.16%.

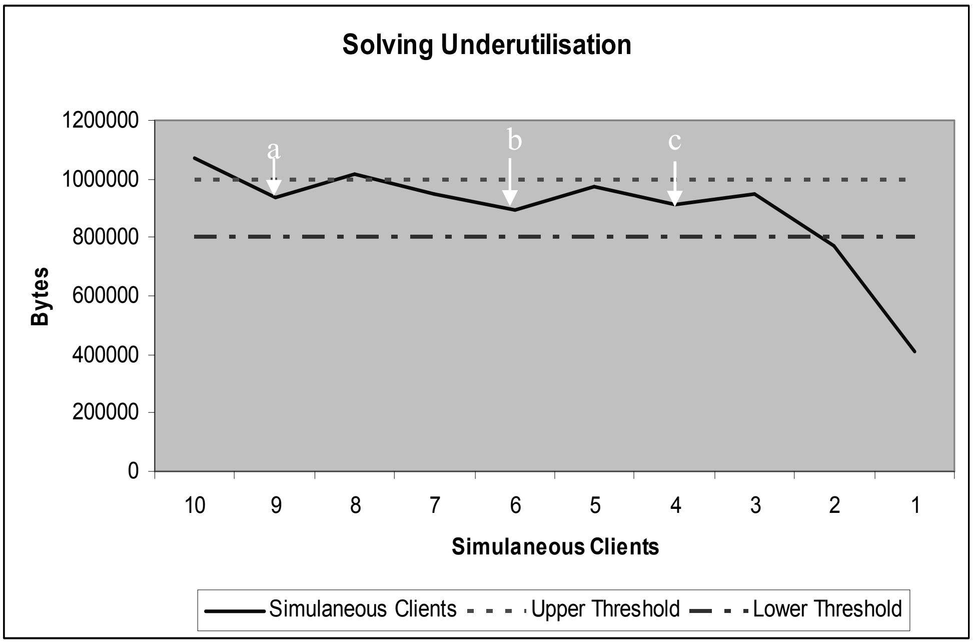


Figure 5-13. Responsive content adaptation for solving the underutilization problem. Tag extraction is applied to every n^th^ document in order to maintain the desired level of network usage. The value of n is increased as the network usage falls.

Responsive adaptation was shown to be capable of maintaining a specified level of network usage, and to avoid the under utilization problem described by Abdelzaher and Bhatti (1999). The responsive adaptation approach implemented makes use of tag extraction, which completely removes content from within documents and may not be desirable. However, the system is intended as a proof-of-concept implementation, and tag extraction could be replaced by another adaptation module that would produce more desirable results. For example, reducing image and video quality would reduce network load while still maintaining the document content.

5.4. Summary

Content adaptation techniques reformat HTML and XML documents, transforming the layout to be better suited for display on small-screen devices. Personalisation provides functionality for clients to specify their adaptation preferences, allowing users to customise content documents. The network management module adjusts adaptation settings according to network usage levels, freeing up network resources. However, the approaches should not detract from the overall user experience by adding a large processing overhead to the response time of the proxy server, and thus experimentation focussed on analysing the performance of the adaptation modules and extensions implemented. All of the three adaptation modules indicated a linear relationship between the size of the content document, and the processing time required. Figure 5-14 provides a comparison of the performance of the implemented modules, showing the average percentage increase created by the adaptation modules.

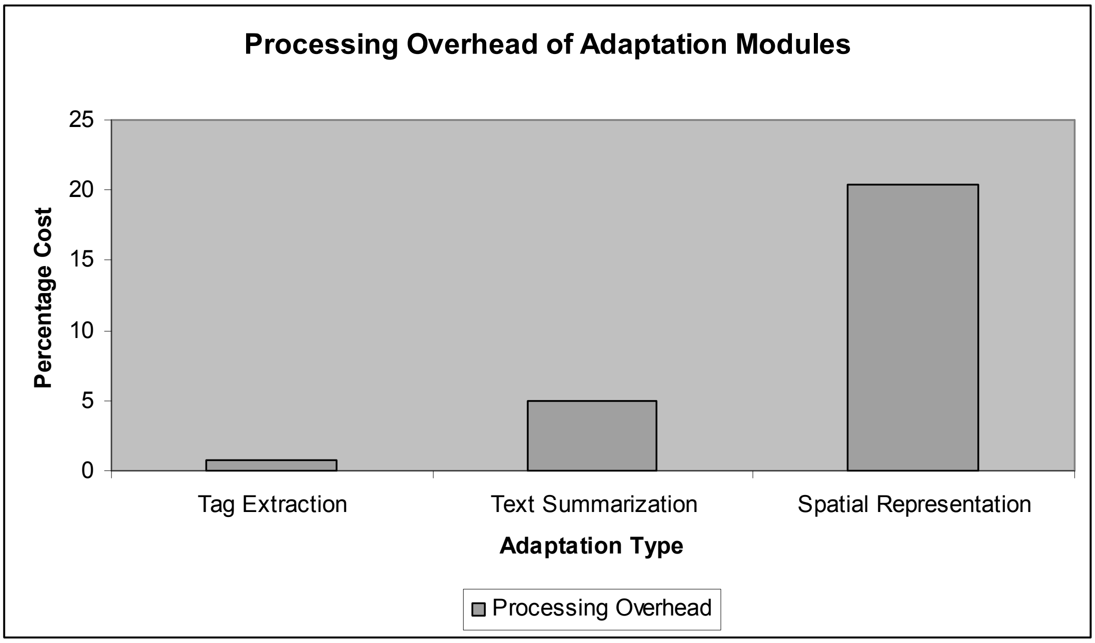


Figure 5-14. A comparison of the performance of the module implementations, showing the additional cost incurred by implementing each module.

Experimentation showed the tag extraction module to function correctly for all possible well formed HTML and XML structures, whereas the text summarization and spatial representation modules are limited to a subset of HTML and XML document structure and require the use of naming conventions. The time taken for the tag extraction module added a processing overhead of 0.72% to the response time of retrieving the sample HTML document, text summarization added 4.98%, and spatial representation 20.42% (Figure 5-14). While these increases appear to be drastic, the resulting actual increase in the round trip time is minimal, and indicates that the performance of the modules is acceptable. The personalisation extension was implemented successfully for devices running IE, providing a method of specify adaptation preferences while adding an average overhead of 5.96% to the time taken to retrieve the sample document. Responsive adaptation was shown to successfully reduce network traffic by utilising the tag extraction module, and provided a mechanism to solve the underutilisation problem. The average processing overhead of the extension was 0.16%. The next Chapter details the implementation of three digital electronic display systems, which were designed in order to evaluate the adaptation framework with regard to the aims detailed in Chapter 3, Section 3.2.

Chapter 6

FRAMEWORK EVALUATION

This Chapter describes three implementations based on the responsive adaptation framework presented in Chapter 3. The systems were designed in order to evaluate the framework according to the criteria established in Chapter 3, Section 3.2. For each system, an electronic display system was developed, and then extended according to the proposed framework by implementing the modules and extensions described in Chapter 4. The first system is a digital slide show presentation system, which is modelled around educational and corporate presentations, and provides a basis to evaluate the framework for an application consisting of only a single display terminal. The second system is a digital signage system, designed around the generalised digital signage architecture defined in Chapter 3, Section 3.1, and provides a platform to evaluate the framework in a more extensive implementation. This system was implemented as a means to evaluate the effectiveness of the framework for digital signage approaches, which is crucial owing to the large number and wide range of digital signage systems in existence. The first two implementations provide a controlled domain in which to evaluate the proposed adaptation framework. In both instances, content documents are developed specifically for the applications, and thus adhere to a set structure and naming conventions. The digital slide show system is extended using the text summarisation and spatial representation modules, and the digital signage approach is extended using the tag extraction and spatial representation modules. In contrast to these approaches, the third system provides internet access, and is designed to test the framework for a wide variety of content documents. This approach implements the tag extraction module, the personalisation extension, and the network management extension. Different programming languages are used for the systems, in order to evaluate the potential of the framework to be implemented in different development environments. Each electronic display system implementation provides wireless access through an 802.11g wireless access point.

6.1. Digital Slide Show Presentations

Rapid advances in mobile device technology have resulted in an increased adoption of these devices in the corporate sector and, more recently, in the educational sector (Finn & Vandenham, 2004). In addition, mobile technology has opened up new opportunities to create novel learning experiences (Rogers *et al.*, 2004). While traditional computer laboratories offer occasional exposure to learning technology, mobile devices now provide the opportunity for frequent exposure making them an integral part of the learning process (Roschelle, 2003). Mobile devices have also been shown to significantly increase students' ability to manage their learning environment and to collect information (Zurita & Nussbaum, 2004). Formal lecture sessions continue to be a popular method of presenting information to university students, and, over the last ten years, digital slide show presentations have become the standard medium of delivery (Bartsch & Cobern, 2003). Students are often encouraged to make study notes during lectures. They then may or may not have access to the original presentation slides after the lecture. Common presentation software packages, such as Microsoft PowerPoint, have built-in web publishing features which make it easy to convert slide shows to HTML format. This, together with numerous content extraction approaches for web content (Alam & Rahman 2003; Buyukkokten *et al.* 2001; Chen *et al.* 2003; Han *et al.* 2000; Kunze & Rosner 2001; Gupta *et al.* 2003; Gupta *et al.* 2004), and ubiquitous mobile computing has provided an opportunity for the novel integration of these technologies for use in the teaching environment. A system that applies content extraction techniques to digital slide show presentations, and provides a mechanism for annotating selected content and for storing these notes is presented. This system illustrates the use of mobile devices as a support tool in traditional learning environments. The main role of the system is therefore to augment learning, and not to deliver instructional content.

The standard architecture for delivering electronic presentations incorporates a large display screen, whereby the displayed content is controlled by a desktop or laptop computer, and the presentation system extends this approach according to the adaptation framework, providing wireless access to mobile devices (Figure 6-1). This implementation is designed to evaluate the flexibility of the framework by applying the approach to an alternative field of electronic display systems. The extensibility of the framework is tested by implementing an extension which allows for note taking. The server hosts the content adaptation system, implemented as a mobile web application, and the actual presentation content in HTML format. Mobile device users connect to the server through the wireless AP and are provided access to the lecture content through the adaptation system (Figure 6-1). A converter module automatically converts digital slide shows into HTML format, and generates additional information required by the spatial representation algorithm. The slide show application allows mobile device users to navigate and annotate selected content. Figure 6-2 provides an overview of the object model of the approach. The slide show application was implemented in order to test the robustness and effectiveness of the text summarization and spatial representation modules in a limited domain of input documents. Usability was not a consideration in the evaluation of the approach.

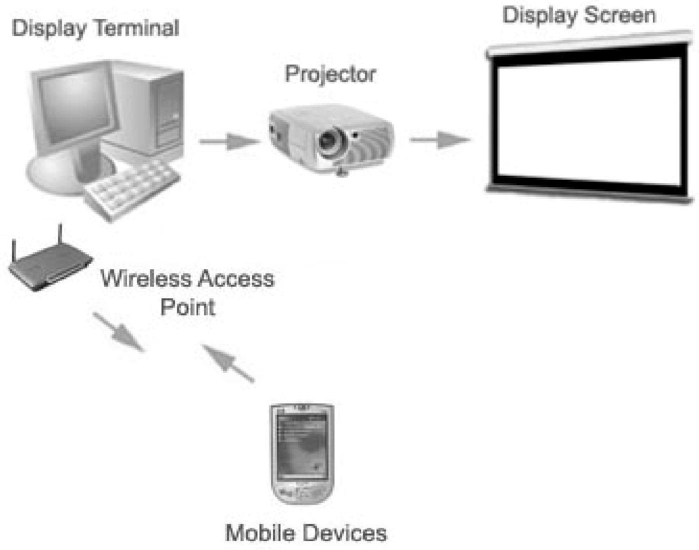


Figure 6-1. High-level architectural view of the slideshow content adaptation system.

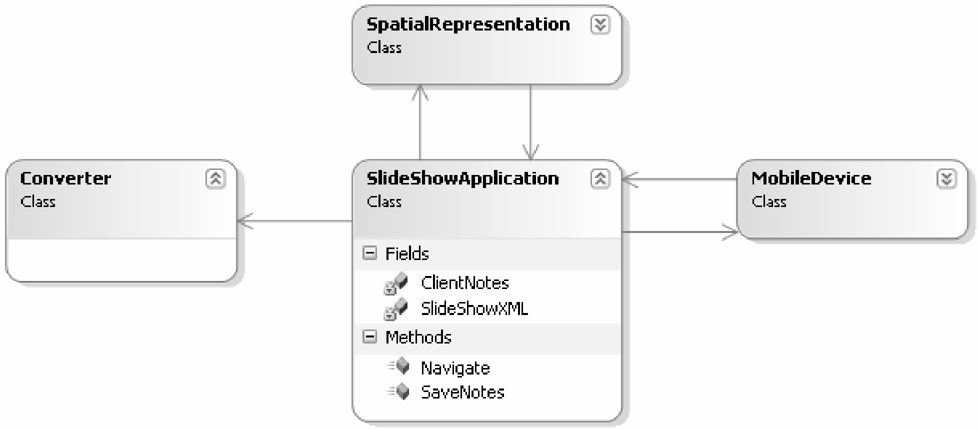


Figure 6-2. Object model of the digital slide show application.

West *et al.* (2005) have provided an implementation of this approach for use with lecture presentations.

6.1.1 Adaptation of Slide Show Presentations

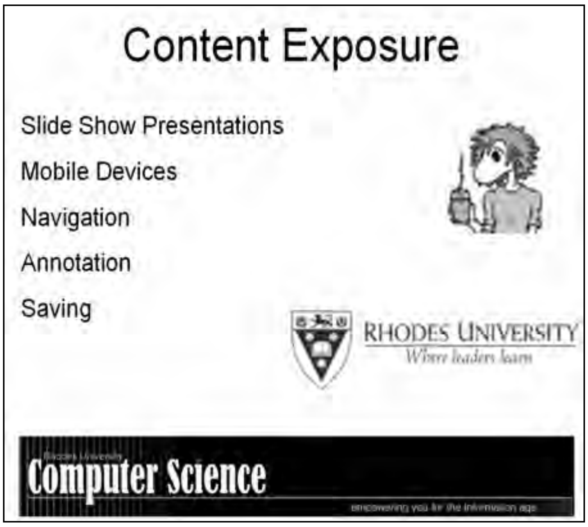
In order to apply standard content adaptation techniques it is necessary to convert slide show files into a markup language. Slide show presentation software, such as Open Office, converts each slide into a single image which is then displayed in an HTML document. This makes identifying content within a slide complex since it would be necessary to analyze the image. However, other presentation software, such as Microsoft PowerPoint, maintains the content from the original slide during the conversion process and represents each content object separately, thus making content adaptation a simpler process. In addition, PowerPoint is supplied with an API for use with the .NET framework, which exposes certain functionality and allows certain processes to be automated. PowerPoint was therefore the presentation software selected as a base format for creating and storing digital slide show presentations for use with the content adaptation system.

Conversion from PowerPoint file format to HTML is automated by utilizing the PowerPoint API in a C# windows application. Conversion is carried out in code by loading the desired PowerPoint file, then executing the “Save As” command and specifying HTML version 3 as the output type. The C# application will search a specified directory for previously unconverted PowerPoint files at a set time interval, and automatically convert any appropriate files discovered. After conversion the necessary content adaptation techniques are applied to the generated HTML file. An XML file is also created during the spatial representation process, and this is used to store a detailed description of each slide in the presentation, including: bound box coordinates for each content block, the names and locations of all images and the actual HTML content for each block. The bound box coordinates are necessary to create the image map using the image generated during the spatial representation process. When a user clicks on an area of the image the screen coordinates are captured and are matched against the bound box coordinates stored in the XML file, locating the corresponding content block.

6.1.2 Mobile Web Application

The client-side interface was implemented as a Microsoft ASP.NET mobile web application, which has the benefit of requiring no smart client application, only a web browser. A further advantage is that the majority of the application processing is carried out on the server-side, reducing the load for the limited processing power of the mobile device (Hadjisotiriou *et al*. 2004).

The initial application screen provides the user with mechanisms to navigate through the slideshow or to navigate within a single slide (Figure 6-4a). The spatial representation shown is derived from the sample content document pictured in Figure 6-3. This document was created as a test document in Microsoft PowerPoint, and contained text and image objects, which are typical content types used in traditional lecture situations. The “Next” and “Previous” buttons allow for navigation to the desired slide. A label indicates the index of the current slide, and corresponds to the actual slide being shown on the large display screen (which also bears an index number). This assists users to follow the slide show presentation and to navigate to the currently displayed slide. Information that is desired can be selected, and the content is displayed in the annotation screen (Figure 6-4b). Annotations can be added using the textbox. The “Add” button temporarily stores annotations and the corresponding content on the server before returning the user to the previous screen. The “Cancel” button will return the user to the active slide without storing annotations or content. Generally, selecting a content block from the spatial representation image map of the slide will navigate the user to the annotation screen (Figure 6-5a). However, it is also possible to view and annotate multiple content blocks by checking the “Multiple” checkbox. This prevents navigation to the annotation screen until the user selects the “Annotate Selected” button (Figure 6-5b). The “Save” button prompts the user to save all annotations and selected content. Notes are saved as a Rich Text Format (RTF) file. RTF was developed as a format that would be readable by all common word processing applications, and can be viewed on most operating systems. RTF is capable of storing images and basic formatting information. Macros can not be inserted into RTF files, and the format is thus not susceptible to macro viruses. RTF files require less memory than other word processing formats, making the format useful for web downloads and desirable for mobile devices owing to their relatively small memory capacity (Senior Corps Tech Centre, 2005). Figure 6-6 provides an example of generated lecture notes, viewed using Microsoft Word.



Content Exposure

Slide Show Presentations

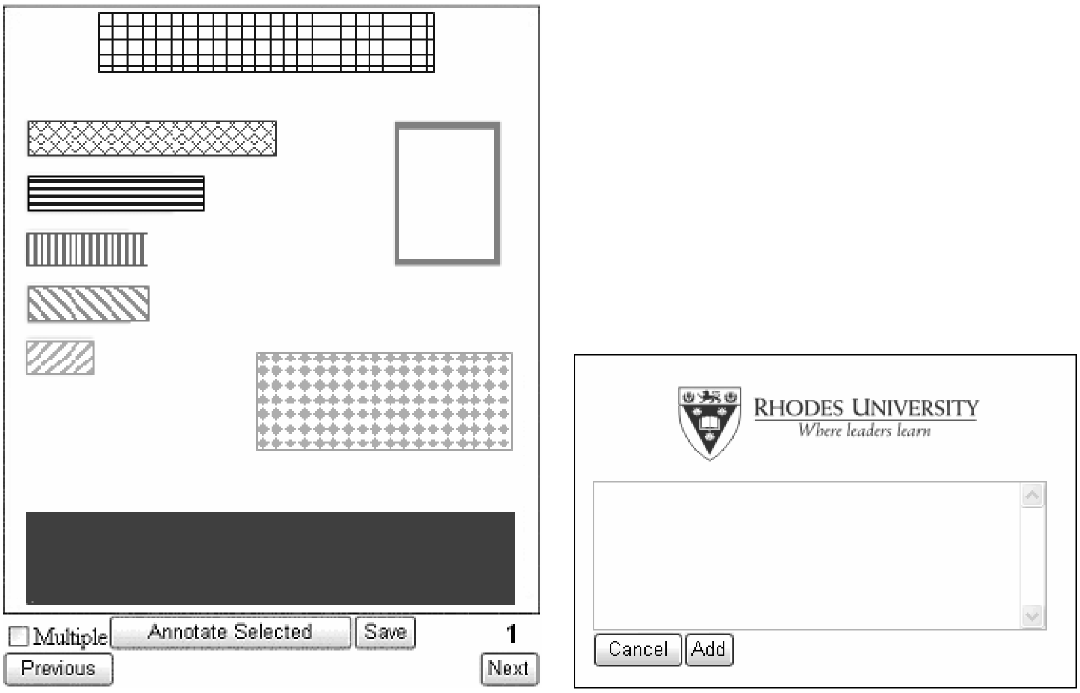
Mobile Devices

Navigation

Annotation

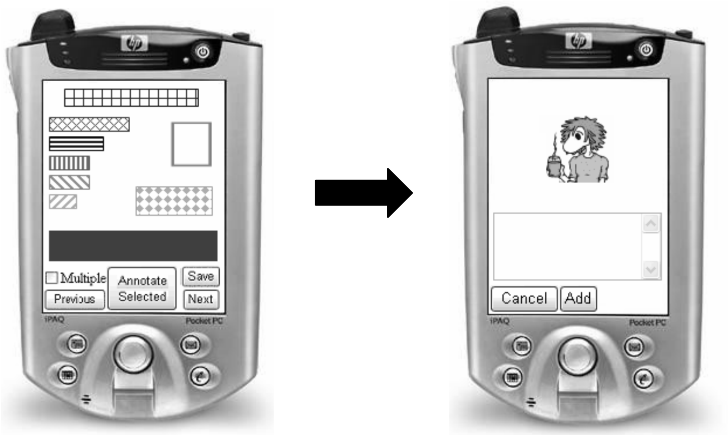
Saving

Figure 6-3. Sample content document.

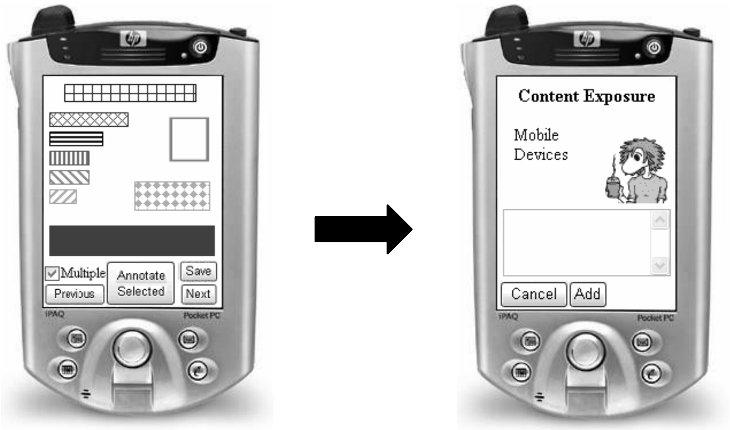


(a) (b)

Figure 6-4. The user interface of the mobile web application. (a) The initial display screen, showing spatial representation. (b) The content viewing and annotation screen resulting from selecting a content object.

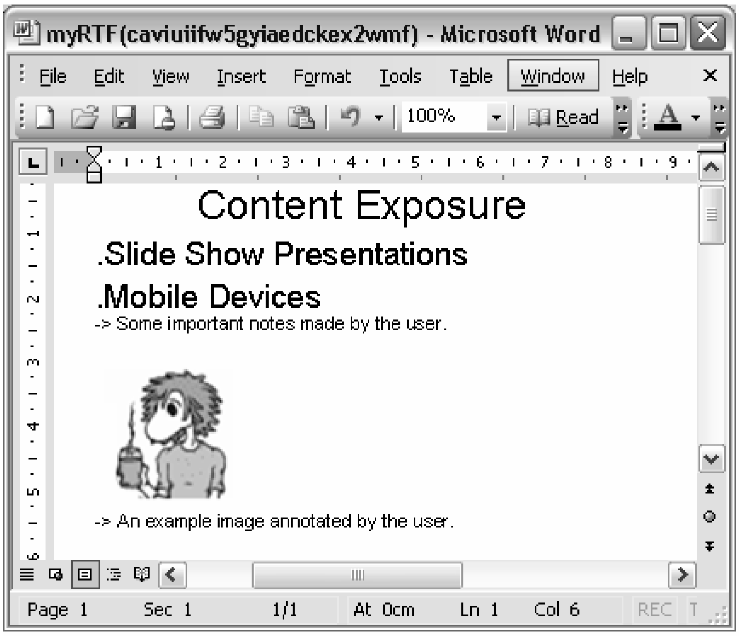


(a)



(b)

Figure 6-5. Using the image map to display selected content. The representation corresponds to the slide shown in Figure 3a. (a) The results of selecting a single content block. (b) The results of using multiple select to select numerous content blocks.


myRTF(caviuiifw5gyiaedckex2wmf) - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%
Read
Page 1 Sec 1 1/1 At 0cm Ln 1 Col 6 REC

Content Exposure

.Slide Show Presentations

.Mobile Devices

-> Some important notes made by the user.

-> An example image annotated by the user.

Figure 6-6. Lecture notes created using the content adaptation system, viewed using Microsoft Word.

6.1.3 Framework Extensions

The presentation system provides a simple implementation, consisting of a single display terminal, with which to evaluate the adaptation framework and to test the spatial representation and text summarization modules in an environment with standardised XML structure and naming conventions. This implementation is compared to the requirements of the adaptation framework (Chapter 3, Section 3.2) in order to evaluate the success of applying the framework to this approach.

- Wireless access – This is provided through the incorporation of an 802.11g wireless access point.
- Content adaptation – A spatial representation module and a text summarisation module divide content documents into segments, providing a method of navigation and reducing the dimensions of a document.
- Modularity and extensibility – The extensibility of the framework is accented through the implementation of a note taking extension.
- Remove complexity from clients and servers – The digital slide show application automatically transforms PowerPoint slide shows into HTML documents, and provides access to this content through a web application. A spatial representation algorithm is used to modify the content documents. The use of a web application as middleware means that the only client-side software required is a web browser.
- A generalised, flexible framework – This system makes use of PowerPoint documents owing to the facilities available to convert these documents into HTML format. However, the framework could be applied to any presentation system that provides for the representation of slide show presentations in a mark-up language document.
- Personalisation – No personalisation approach was implemented.
- Responding to network conditions – No network management approach was implemented.
- Low response time overhead – Evaluation of the spatial representation and text summarisation modules in Chapter 5 indicated that the modules provided an acceptable level of performance.

6.2. Digital Signage

Digital signage networks are being readily adopted in a wide range of environments, including the corporate and educational sectors, retail outlets, and public places such as airports, hotels, and sports stadiums. The large number, and wide range, of digital signage networks makes this form of electronic display network an important approach. For this reason, an XML-based digital signage system was developed that allows for the control of a digital signage network, and is extended according to the adaptation framework by applying the tag extraction and spatial representation modules. The system forms a basis for evaluating the effectiveness of the framework when implemented in digital signage systems. Figure 6-7 illustrates an overview of the physical system architecture. Content display layouts, for the display terminals, are designed on a mobile device or remote computer and are uploaded to the central server, which then updates the display terminals. Content is stored either locally on each display terminal or remotely on content servers. A wireless access point is attached to each display terminal to allow mobile device access to the content shown on screen. Each display terminal runs a proxy server to facilitate user access to the content, and to provide content adaptation capabilities. Figure 6-8 illustrates the object model of the digital signage system. A client requests a content document from the web application using their mobile device. In turn, the web application retrieves the relevant document from a content server, which is then transformed using the spatial representation and/or tag extraction module according to the adaptation settings specified by the system administrator. The web application, tag extraction, and spatial representation modules, extend the digital signage system in terms of the adaptation framework.

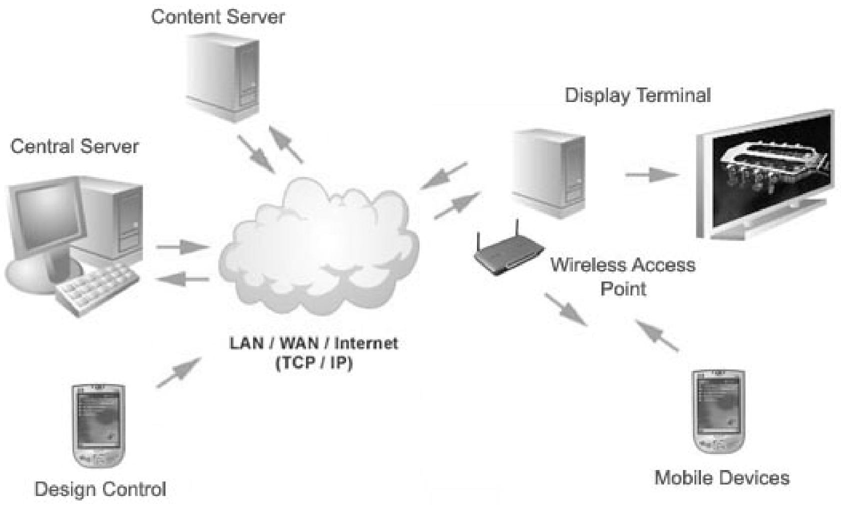


Figure 6-7. Digital signage system extended to include wireless access and content adaptation.

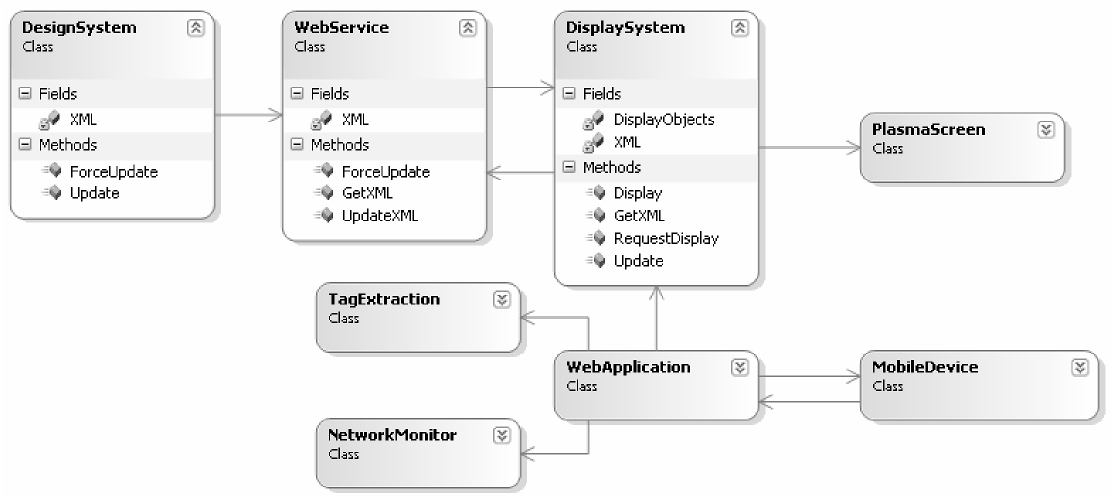
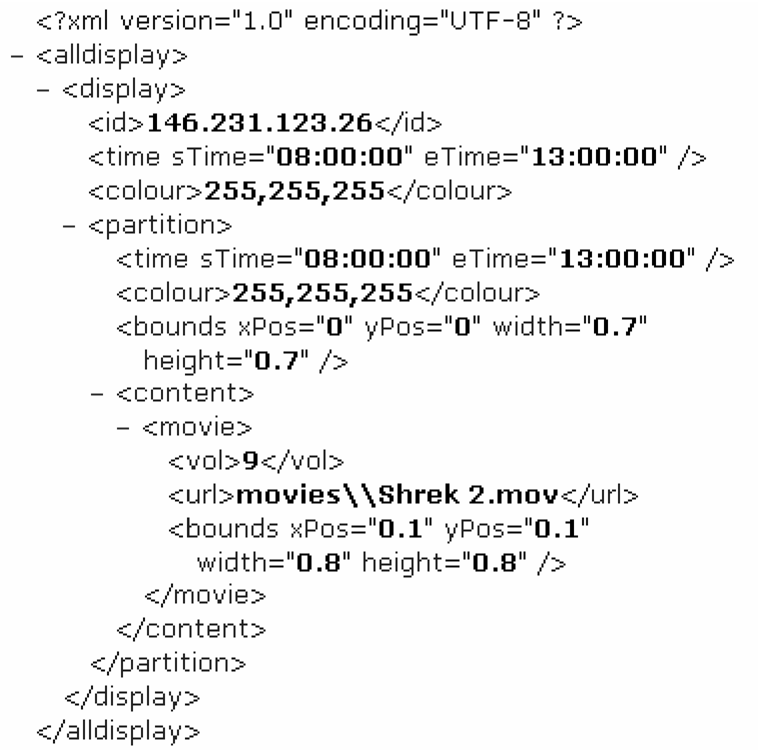


Figure 6-8. Overview of the digital signage system architecture.

6.2.1 Content Display Document

A display screen for the digital signage system consists of a base panel, which in turn can contain an unlimited number of content panels which contain a combination of text, images, and video content objects. The display layouts are described by an XML document (Figure 6-9). The display document consists of one of more <display> tags, which define a display for a particular terminal or set of terminals identified by the <id> tag. Each display is comprised of an unlimited number of partitions, has a start time and end time, and a background colour. In turn, each partition has an associated start and end time, a background colour, and a set of coordinates which define the partition bound box. The partitions start time may be later than the display start time, and the end time may be earlier than the display end time. This allows content panels to be added or removed during a displays life time, providing greater flexibility to the system. Each partition contains an unlimited number of content objects which can be text, an image, or a movie. A text object contains the actual text to be displayed, font, text size, text colour, and the bound box coordinates. An image object contains a URL describing the location of the image, and the bound box coordinates. While, a movie object contains the volume of the movie, on a scale from 0 to 10, a URL describing the location of the movie, and the bound box coordinates.



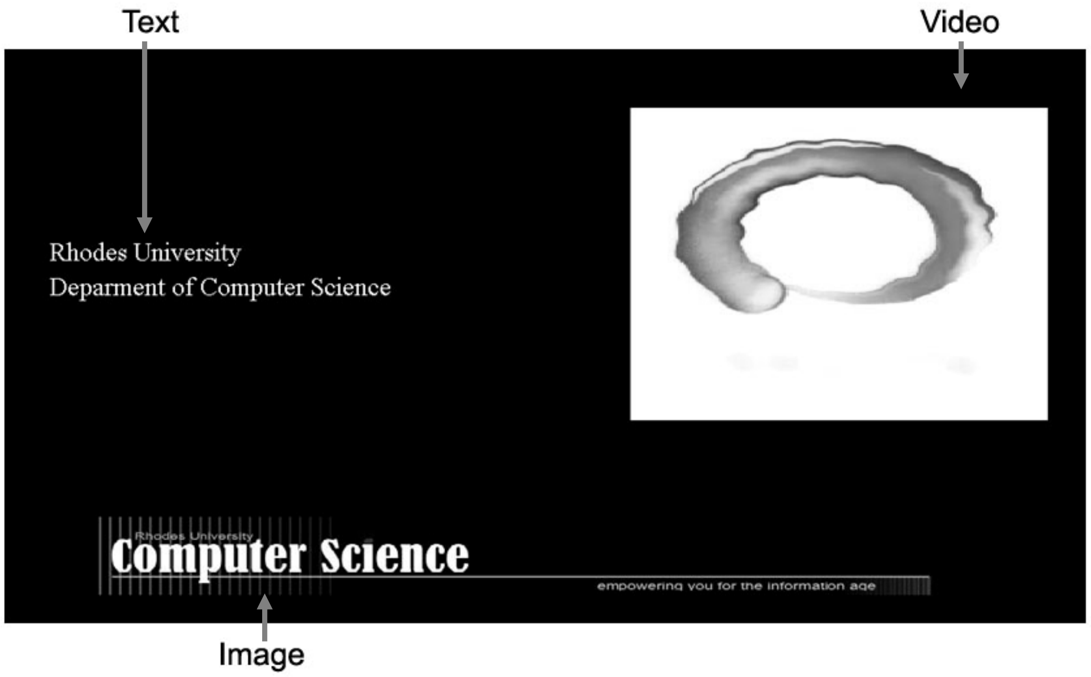
```
  <?xml version="1.0" encoding="UTF-8" ?>
- <alldisplay>
  - <display>
      <id>146.231.123.26</id>
      <time sTime="08:00:00" eTime="13:00:00" />
      <colour>255,255,255</colour>
    - <partition>
        <time sTime="08:00:00" eTime="13:00:00" />
        <colour>255,255,255</colour>
        <bounds xPos="0" yPos="0" width="0.7"
          height="0.7" />
      - <content>
        - <movie>
            <vol>9</vol>
            <url>movies\\Shrek 2.mov</url>
            <bounds xPos="0.1" yPos="0.1"
              width="0.8" height="0.8" />
          </movie>
        </content>
      </partition>
    </display>
  </alldisplay>
```

Figure 6-9. Example XML document describing a display.

6.2.2 Display Terminal

Each display terminal computer runs a Java application that controls the content shown on-screen, and handles the retrieving of the content documents. Figure 6-10 shows an example display screen created by the application. The display contains three partitions, each containing a single content object. These are: a text object, an image object, and a video object. QuickTime for Java (QtJava)[xxxiv] is a Java class library that allows QuickTime content to be displayed in Java applications, and was implemented to provide the display terminals with the ability to display multimedia content. XML parsing was done using the Java API for XML processing (JAXP). Both a SAX and a DOM parser were implemented for comparison purposes.

[xxxiv] http://developer.apple.com/quicktime/qtjava/



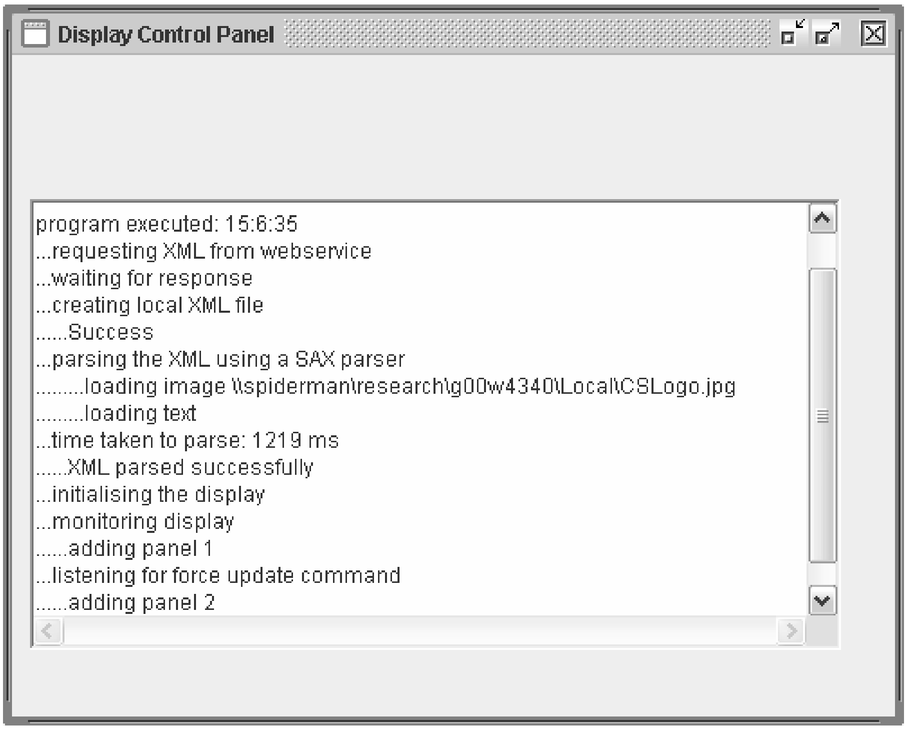
Rhodes University
Deparment of Computer Science

Figure 6-10. An example rendered display, containing text, an image, and a video object.

On start-up, the application will request the relevant XML from the web service contained on the central server. Messages are sent using the Simple Object Access Protocol (SOAP)[xxxv], which is a platform and language independent communications protocol based on XML. All messages in the digital signage network are sent using SOAP, meaning that different modules need not be created with the same programming language, or even run on the same operating system. The retrieved XML is then saved locally on the terminal, and is parsed to build up the display and check for errors. All events and errors are reported on the display control panel (Figure 6-11). Each display may contain an unlimited number of panels. Each panel can contain multiple video, images and text. The base panel is implemented as a Java Frame[xxxvi] object. Java JPanels[xxxvii] are used to represent the partitions. Content objects are added to the JPanel as JComponets[xxxviii]. This is because QTJava components can be cast into JComponent objects, allowing QuickTime multimedia content to be added to a Java window. A null layout manager was used to enable absolute positioning of content objects and panels. Once the display is running, the application continuously listens for a force update command from the central server.

[xxxv] http://www.w3.org/TR/soap12-part0/
[xxxvi] http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Frame.html
[xxxvii] http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JPanel.html
[xxxviii] http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JComponent.html

If received, the display application restarts, requesting the new XML document. The display is further monitored in order to identify panels that need to be added or removed. Video objects loop continuously until they are removed from the display. When the display end time has been reached, the application closes the display and requests a new XML display document.



```
program executed: 15:6:35
...requesting XML from webservice
...waiting for response
...creating local XML file
......Success
...parsing the XML using a SAX parser
.........loading image \\spiderman\research\g00w4340\Local\CSLogo.jpg
.........loading text
...time taken to parse: 1219 ms
......XML parsed successfully
...initialising the display
...monitoring display
......adding panel 1
...listening for force update command
......adding panel 2
```

Figure 6-11. The display terminal control panel, displays status information and performs error reporting.

6.2.3 Content Display Server Application

The server application stores an XML document containing display details for all existing display terminals, and supplies this XML to the relevant terminals on request. The application is implemented as a web service, created with Visual Basic .NET (VB.NET). The use of VB.NET further emphasises the modularity of this system, illustrating the use of different programming languages and development environments. Implementing the server as a web service provides flexibility, since the display terminal applications can be replaced by an application written using any language that supports SOAP.

The `GetXML` method allows the XML content display file to be uploaded onto the server Figure 6-8. The file contains the displays for all the display terminals, and is stored locally. Display documents are requested by display terminal applications by calling the `GetDisplay` method, which takes an IP address as the only parameter. The IP address is used to locate the relevant XML within the original document, which is then extracted and returned to the requesting display terminal application. The IP address is then added to a list of active display terminals. If the requesting terminal IP address does not match any in the display document, an error is returned. The `GetIDs` method returns the list of IP addresses for all known display terminals. The `ForceUpdate` method forces a specific display terminal to update the XML display document and refresh the actual display. The method takes an IP address as a parameter, and is called from a remote location. This method is used when an error has been made in the design of a display document, or when a document has been modified or cancelled.

6.2.4 Content Display Design System

Opie (2004) developed a web application that facilitates the design of content displays, and provides functionality for updating the central server via the web service. The web application was developed using ASP.NET, and can be accessed by clients running both IE and Pocket Internet Explorer (PIE) meaning that the application can run on both desktop computers and mobile devices running Microsoft Windows based operating systems. A display is created by inputting information using the user interface, including the building of partitions and specifying the location of the content. Once this process is completed, the application generates an XML document based on the information given. This XML document is then sent to the web service using SOAP. The web application allows the client to send `ForceUpdate` commands to the server, providing control over the digital signage network.

6.2.5 Framework Extensions

The aim of the digital signage application is to provide an example signage system, and to extend this system according to the framework provided. This implementation provides a platform for testing the tag extraction and spatial representation modules in a controlled environment with a specific XML structure. An ASP.NET mobile web application provides access to the content for mobile device users, acting as the proxy server in the framework. The system incorporates a tag extraction module and a spatial representation module, which were implemented as detailed in Chapter 4. Implementing the framework extends the digital signage system to provide wireless access via mobile device to the content displayed. Content adaptation modules provide a mechanism to transform content to display more appropriately on small-screen devices, and to reduce the level of network usage. A comparison of this implementation, to that of the requirements of the adaptation framework (Chapter 3, Section 3.2) is provided below.

- Wireless access – This is provided through an 802.11g wireless access point.
- Content adaptation – The tag extraction and spatial representation modules provide content adaptation. Tag extraction provides a means to reduce network load by removing rich content such as video. Spatial representation provides a convenient means to navigate the display.
- Modularity and extensibility - The framework implementation is modular and extensible in terms of the content adaptation modules and extensions, which can be added or removed. However, these modules and extensions are closely coupled with the web application, meaning that removing or replacing this module would remove all the modules and extensions implemented.
- Remove complexity from clients and servers – A mobile web application provides an HTML version of the content display, meaning that the client device requires only a web browser to view the content. The web application retrieves the documents from the content servers, and transforms these documents using the content adaptation modules. Thus the web application handles all framework complexity.
- A generalised, flexible framework – The framework implementation is applicable to any digital signage system that utilises XML documents as the format for describing display layouts. The spatial representation module restricts the digital signage system to a consistent document layout.
- Personalisation – No personalisation approach was implemented.

- Responding to network conditions – No network management approach was implemented.
- Low response time overhead – The results obtained from testing the tag extraction and spatial representation modules in Chapter 5 show the response time of the implementations to be minimal, making this a feasible approach in terms of performance.

6.3. A Content Adaptation System for Web Browsing

The digital signage system and the digital slide show system handle content documents of a pre-defined structure. That is, the structure of the XML and HTML documents dealt with is restricted in terms of layout and naming conventions of tags, and thus all instances of the structural domain can be designed for. An adaptation system for web browsing was implemented in order to test the adaptation modules in an environment where the structure of documents is not pre-defined, and thus the intended layout of the documents is unknown. Although the system is not an electronic display system, the goal of this implementation is to evaluate the content adaptation modules and extensions to the framework in an environment where the structure of content documents is undefined, in order to determine the robustness of the modules developed. The web browsing system is further extended to include the personalisation approach and the network management implementation. Crunch 2.0 was implemented as the framework proxy server (Chapter 2, Section 2.2.5). The Crunch platform is a thread-based proxy that facilitates the implementation of content adaptation plug-ins (Gupta *et al.* 2004). Crunch is an open source proxy and can thus be extended to include the personalisation and the network monitoring modules. Figure 6-12 provides an overview of the web browsing system architecture. Requests from mobile devices are received by the Crunch application, which then retrieves the relevant document from the network and performs any required adaptation functions before returning the document to the client. Figure 6-13 shows the object model of the web adaptation system. A client requests a content document from the internet, or local network, through the Crunch proxy server. In turn, Crunch retrieves the requested document, and parses the mark-up into a DOM tree representation which is then provided to all the adaptation plug-ins and modified accordingly. The resulting document is then returned to the client. The personalisation extension developed for

Crunch stores client adaptation preferences in a local database. A browser helper object (BHO), which resides on the client mobile device, provides the user with the functionality to transmit user preferences to the proxy server. A BHO is an extension to IE, and allows for the monitoring and manipulation of web requests. The network management extension monitors network usage levels, and adjusts tag extraction settings to reduce network load.

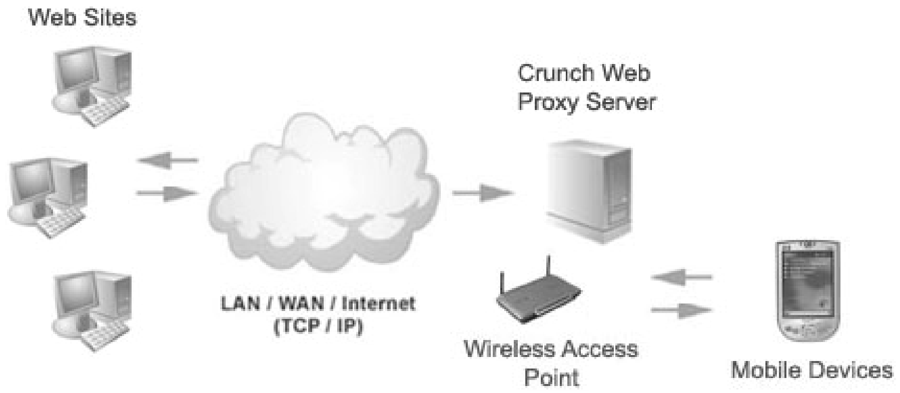


Figure 6-12. Overview of the system architecture of the content adaptation system for web browsing.

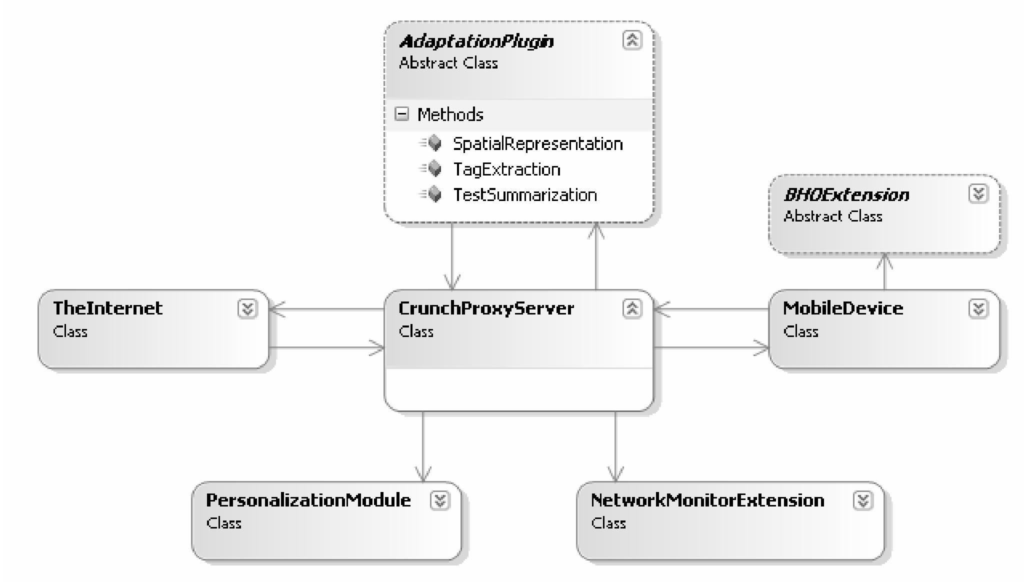


Figure 6-13. Object model of the web content adaptation system.

6.3.1 Framework Extensions

The Crunch system receives web requests from clients, retrieves the relevant document, and creates a DOM representation of the document. Crunch plug-ins extend the `ProxyFilter` abstract class, which forces the plug-in modules to implement a method which handles the processing of the document. Each module receives three DOM representations of the document. The first is a copy of the document manipulated by previous plug-ins, and is used for rollback purposes in case errors are encountered. The second is the document to be processed. The third is a copy of the original document received by the proxy, and can be used for quality checking. Once processing on the document has been completed, the plug-in returns the modified DOM tree.

The personalization module was implemented as described in Chapter 4. A BHO on the client device appends the user's adaptation preferences to web requests. Settings are stored locally on the client side and can be altered at any point. The Crunch proxy server was modified by implementing a function to identify appended user settings. Preferences are removed from the request, which is then returned and handled as normal by the proxy server. The function stores the extracted preferences, along with the client IP address, in a MySQL database. The tag extraction module was then altered to request client preferences from the database to obtain adaptation settings. Adaptation preferences included:

- Tag Extraction – A selection of either True (1) or False (0) for embedded objects, images, and scripts.

Thus a request containing personalisation settings from a client would take the form: webaddress*settings, where webaddress refers to the original web request, and settings refers to the adaptation settings in the form 1/0;1/0. For example, settings of 10 would translate to tag extraction of embedded objects, and no extraction of images. The '*' is used to identify that adaptation settings have been appended by the client

The network monitoring system was also implemented as detailed in Chapter 4, and adheres to the same logic as detailed in Section 4.3.2. The responsive system is utilised by the tag extraction module as tag extraction allows greater flexibility in reducing document size, providing a large permutation of adaptation settings. The success of the implementation is compared to the requirements of the responsive adaptation framework (Chapter 3, Section 3.2):

- Wireless access – This is provided through the installation of an 802.11g wireless access point.
- Content adaptation – Tag extraction provides a method of manipulating content to reduce document dimensions and the size of the document in bytes.
- Modularity and extensibility – The framework is extended to include a personalisation approach and a network management system.
- Remove complexity from clients and servers – The use of the Crunch proxy server provides a middleware application which can automatically modify content. The personalisation approach implemented requires the use of a client-side BHO, and thus requires increase the complexity on the client-side.
- A generalised, flexible framework – The framework was successfully applied to a generalised system for web browsing.
- Personalisation – A personalisation module was implemented that allows clients to specify their adaptation preferences. These settings are stored on the server-side and applied to all requests made by the specific client.
- Responding to network conditions – A network management approach was implemented that monitors network usage levels, and adjusts the tag extraction module settings in order to reduce network load. The approach successfully combats the underutilisation problem described in Chapter 2, Section 2.4.2.
- Low response time overhead – The results described in Chapter 5 indicate that the performance costs of implementing the tag extraction module, personalisation extension, and the network management extension, are acceptable. However, these results measure the cost of using a single module or extension, and not the cost of using these implementations together. Thus the total cost could detract from the benefits gained.

6.4. Summary

Three electronic display system implementations are described in this Chapter. These systems were developed as a range of platforms to test the proposed responsive adaptation framework. Each system was extended according to the framework, and each implementation was designed to test different aspects of the approach. The digital slide show application allows mobile access to Microsoft PowerPoint slides through a mobile web application running on a web browser, and is extended by implementing the text summarization and spatial representation algorithms. A further extension provides the functionality to make notes, and is designed to test the extensibility of the framework. This approach provides an electronic display system consisting of a single display terminal, which is used as a platform to test the text summarization and spatial representation modules in a domain. The digital signage system is an XML-based signage network, providing a design module, a central server module, and a display terminal module. This system is then extended, according to the framework, by adding a proxy server, tag extraction module, and a spatial representation module. The approach provides an example digital signage system, and a platform to test the tag extraction and spatial representation modules in an environment where the content document structure is pre-defined. A content adaptation system has been implemented for web browsing. Crunch 2.0 is used as the proxy server, as it is an open source implementation and provides for the development of content adaptation plug-ins. The tag extraction module was modified to function as a plug-in for the Crunch proxy server. Crunch was further extended by implementing a personalization approach, and a responsive approach for use with the tag extraction module. These framework implementations provide the base platform for evaluating the adaptation framework. The next Chapter provides a review of the study, details the contributions made and the conclusions reached, and provides a discussion on possible future research.

Chapter 7

CONCLUSIONS AND FUTURE RESEARCH

7.1. Review

This work aimed to develop a framework for implementing a responsive adaptation system in electronic display networks. Previous content adaptation approaches, and adaptation systems, were investigated in order to identify viable techniques. Work related to content personalisation and network management was also investigated. A framework was proposed based on the information obtained from reviewing previous implementations. Content adaptation modules and extensions were developed in order to give substance to the framework, and to provide a platform for assessing the viability of the proposal. Three systems were developed, and then extended using the adaptation framework. This platform formed the basis for evaluating the framework.

7.1.1 The Proposed Framework

The proposed framework aims to fulfil the following criteria (Chapter 3, Section 3.2):

- Wireless access to content for mobile devices.
- Content adaptation to reduce document dimension and size.
- A modular architecture, providing flexibility and extensibility of content adaptation modules and framework extensions.
- A middleware approach, removing the need for complexity on the client and server-sides.
- A generalised, flexible framework, applicable to a wide range of electronic display systems.
- An extension to the framework providing a mechanism for personalisation of adaptation settings.
- A network management extension, providing the ability to monitor network conditions and respond by modifying content adaptation settings in order to reduce bandwidth usage.

- A low response time overhead for all adaptation modules and extensions, so as not to detract from any potential benefits.

Many existing electronic display systems function purely as display terminals, and do not provide a means to access the content shown. Harrison and Andrusiewicz (2004) propose a system for providing mobile device access to content displayed on a digital signage system. However, this approach transfers a predefined set of documents to the requesting device, and does not allow access to the actual content on display. Our proposed framework architecture is based on content adaptation systems, which adopt a middleware approach in order to remove complexity from the client and content server. The framework provides unrestricted access to the content in electronic display systems, and implements content adaptation techniques in order to transform content appropriately for mobile devices. A personalisation extension facilitates user customisation of content documents by allowing clients to specify adaptation preferences. Further, the framework provides a network management extension, which maintains a level of QoS in the network. The framework is limited to electronic display systems which utilise mark-up languages to describe the layout of content documents in the system.

7.1.2 Content Adaptation Modules

The content adaptation modules were implemented to provide a platform for assessing the viability of incorporating content adaptation methods in the proposed framework. Modules were analysed by evaluating the additional response time created owing to processing, and the correctness of the resulting document from the module output. Usability was not a consideration in evaluating the adaptation modules. Three modules were developed: tag extraction, text summarisation, and spatial representation, and were evaluated according to performance, indicating a linear relationship between the size of the content document, and the processing time required. Tag extraction functions correctly for all well formed HTML and XML structures, whereas the text summarization and spatial representation modules are limited to a subset of HTML and XML document structure. The tag extraction module reduces document size, while text summarization and spatial representation reduce document dimension. The time taken for the tag extraction module added a processing overhead of 0.72% to the response time of retrieving the sample HTML document (Chapter 5, Figure 5-1), text summarization added 4.98%, and spatial representation 20.42%. These results indicate that the performance of the modules is acceptable, and does not result in a large processing overhead. Although the processing overhead costs for text summarization and spatial representation appear large (4.98% and 20.42%), the actual cost in time is relatively small (35.85ms and 205.69ms respectively for 128 tags) and can thus be considered negligible.

7.1.3 Framework Extensions

Two framework extensions were developed. A personalisation system provides clients with a mechanism to customise adaptation settings, and a network management system maintains a level of network resource usage. The personalisation extension was implemented successfully for devices running Internet Explorer, and is currently limited to this platform. Users without the personalisation module installed on their devices can still access content, but can not customise content documents. Also, the approach requires users to specify their preferences, and does not provide a mechanism for establishing a user's preference through data mining. The average overhead of retrieving the sample document (Chapter 5, Figure 5-1) increased the response time by 5.96%. Responsive adaptation was shown to successfully reduce network traffic by utilising the tag extraction module, and provided a mechanism to solve the underutilisation problem. The average processing overhead of the extension was 0.16%. This approach makes use of the tag extraction module, which completely removes content objects from documents.

7.1.4 Test Implementations

Three electronic display systems were developed in order to evaluate the proposed framework according to the aims of the framework, established in Chapter 3, Section 3.2. A digital slide show system was implemented to evaluate the feasibility of applying the framework to a presentation system. An XML-based, digital signage system was developed which displayed multimedia content. A generalised system for web browsing was developed to assess the framework in a dynamic environment. The digital slide show system was extended, again using a mobile web application, to provide wireless access, spatial representation, and an extension providing note taking facilities. The digital signage implementation was extended using a mobile web application, and provided tag extraction and spatial representation. The web browsing system was extended using the Crunch proxy server, tag extraction, the personalisation system, and the network management system.

7.2. Summary

The main aim of the work was to extend electronic display networks to provide mobile device access to content. In order to achieve this, a framework was proposed that provides: wireless access, content adaptation, modularity and extensibility, the removal of complexity from clients and servers, a generalised, flexible framework, personalisation, response to network conditions, and a low response time overhead for all modules and extensions. Table 7-1 provides a summary of the results obtained from evaluating the framework according to the established aims, Table 7-2 provides a more detailed summary of the results obtained from testing the performance and effectiveness of the adaptation modules.

Table 7-1. Results of evaluating the success of the framework according to the aims established.

Aims	Achievement	Limitations
Wireless access to content for mobile devices	Facilitated for each framework implementation by adding a wireless access point to every display terminal. Access to content was controlled by a proxy server	None evident
Content adaptation to reduce document dimension and size	A tag extraction module reduces document size. A text summarization module and spatial representation module reduce document dimension	Only tag extraction is not limited to a specific document structure. More detailed benefits and limitations provided in Table 7-2
A modular architecture, providing flexibility and extensibility of content adaptation modules and framework extensions	The framework implementations made use of a varying combination of adaptation modules and extensions	None evident
A middleware approach,	The framework implementations	No single, standard proxy

removing the need for complexity on the client and server-sides	make use of a proxy server to control access to content, and to facilitate content adaptation.	server was implemented. The approach was therefore adapted for use with different servers
A generalised, flexible framework, applicable to a wide range of electronic display systems	The adaptation framework was successfully implemented in three electronic display systems.	Limited by document structure and naming conventions. Modules must be slightly altered for each implementation
An extension to the framework providing a mechanism for personalisation of adaptation settings	A personalisation approach was successfully implemented, and can be customised for use with any adaptation module. Provides negligible response time overhead and document customisation	The approach is currently limited to the IE platform, and requires client-side software. Does not utilise data mining to infer user preferences
A network management extension, providing the ability to monitor network conditions and respond by modifying content adaptation settings in order to reduce bandwidth usage	The responsive adaptation extension was shown to function as intended, reducing traffic on the network while avoiding the underutilisation problem. Provides negligible response time overhead and maintains a set level of network usage	The extension provides a basic mechanism for controlling bandwidth usage, and is limited further by the adaptation module utilised.
A low response time overhead for all adaptation modules and extensions, so as not to detract from any potential benefits	Evaluation of the performance of the modules and extensions indicated that the processing overhead cost was acceptable	The adaptation modules implemented are relatively simple, and do not attempt to infer document structure. More complicated modules could drastically increase processing cost

Table 7-2. Results of experimentation on the framework adaptation modules.

Components	Achievement	Limitations
Tag Extraction	Not limited by document structure Negligible response time overhead Reduces document size and dimension	Removes content
Text Summarization	Negligible response time overhead Reduces document dimension	Limited document structure Does not infer document layout
Spatial Representation	Negligible response time overhead Reduces document dimension	Limited document structure Does not infer document layout

The following conclusions can be drawn from the results of evaluating the components, and the three test implementations of the framework:

- The proposed framework provides a viable solution to providing mobile devices wireless access to content in electronic display networks, as the framework architecture, modules, and extensions successfully address the goals described in Chapter 3, Section 3.2.
- Results from the implementation of three electronic display systems indicate that the framework can be successfully applied to a range of applications.
- The content adaptations implemented provide a viable method of facilitating small-screen device access to content in electronic display networks, as the modules successfully reduce the document dimension and size of documents of a specific structure, and do not introduce a significant response time overhead. Usability of the adaptation modules was not evaluated.
- The personalisation extension successfully provides a method of document customisation while maintaining a negligible response time overhead.
- The network management implementation successfully maintains a specified level of bandwidth usage, and does not adversely affect response time.

7.3. Future Research

Section 7.2 details certain limitations of the proposed framework, and this section provides suggestions for investigating possible solutions to these limitations.

- The adaptation modules developed should be extended to be more complete and more effective. Chen *et al.* (2003) present an approach for detecting web page structure, and such a technique could be implemented in order to make the text summarisation and spatial representation modules more generalised solutions. The tag extraction module currently only replaces content with links, but could be extended to provide more advanced functionality, such as reducing image quality. Alternative content adaptation techniques could be investigated.

- The responsive adaptation module has shown to be a successful implementation in controlling network usage. The current implementation utilises the tag extraction module in order to reduce network traffic, which is a drastic approach since entire content objects are removed from the document. More advanced adaptation modules could be used to provide a less drastic network management approach, and alternative network management approaches could be investigated.
- The personalisation extension allows client devices to specify adaptation preferences. This approach is currently limited to IE, but could be extended to provide plug-ins for other browser applications, such as extensions for Mozilla Firefox[xxxix]. Alternatively, creating a web application for specifying preferences would remove the need for specialised client-side software, allowing any device with a web browser to utilise the personalisation system.
- Currently only three framework extensions have been implemented, personalisation and responsive adaptation. The platform exists to develop further extensions, such as implementing billing modules, and providing product purchasing facilities in the case of digital signage systems.

[xxxix] http://kb.mozillazine.org/Dev_:_Extensions

LIST OF REFERENCES

ABDELZAHER, T. F. & BHATTI, N. 1999, "Web Content Adaptation to Improve Server Overload Behavior", *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 31, pp. 1563-1s577.

ALAM, H., and RAHMAN, F. 2003, “Web Document Manipulation for Small Screen Devices: A Review”. *Second International Workshop on Web Document Analysis (WDA2003),* Edinburgh, August 3.

BARTSCH, R. A., and COBERN, K. M. 2003, “Effectiveness of PowerPoint Presentations in Lectures”. *Journal of Computers and Education*, vol. 41, issue 1, pp. 77 – 86.

BICKMORE, T. W. & SCHILIT, B. N. 1997, "Digestor: Device-Independent Access to the World Wide Web", *Proceedings of the .6th World Wide Web Conference (WWW’06)* pp. 655-663.

BUYUKKOKTEN, O., GARCIA-MOLINA, H. and PAEPCKE, A. 2001, “Text Summarization of Web Pages on Handheld Devices”. *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL),* Pittsburgh, PA, U.S.A., 2-7 June.

CHEN, Y., MA, W.Y., and ZHANG, H.J. 2003, “Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices”. *Proceedings of WWW’03,* Budapest, Hungary, May 20-24.

COMPUTER INDUSTRY ALMANAC, 2005, “Mobile PCs In-Use Surpass 200M”. Available from: http://www.c-i-a.com/pr0605.htm . 2005. [Accessed 2005].

DEGIOANNI, L. 2000, “Development of an Architecture for Packet Capture and Network Traffic Analysis”*,* Graduation Thesis, Politecnico Di Torino, Turin, Italy.

EIRINAKI, M. & VAZIRGIANNIS, M. 2003, "Web mining for web personalization", *ACM Transactions on Internet Technology (TOIT)*, vol. 3, no. 1, pp. 1-27.

ESPOSITO, D., 1999, “Browser Helper Objects: The Browser the Way You Want It”. Microsoft Corporation. 2004. Available from: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp . [Accessed 2005].

FENG, Y. & ZHU, J. 2001, “Wireless Java Programming with J2ME”. Pearson Education, Sam's Publishing. 800 East 96th Street Indianapolis, Indiana 46240, United States of America.

FINN, A., KUSHMERICK, N. and SMYTH, B., 2001, “Fact or Fiction: Content classication for digital libraries”. *Joint DELOS-NSFWorkshop on Personalisation and Recommender Systems inDigital Libraries (Dublin).*

FINN, M., and VANDENHAM, N. 2004, “The Handheld Classroom: Educational Implications of Mobile Computing”. *Australian Journal of Emerging Technologies and Society*, vol. 2, no. 1.

FOX, A., GRIBBLE, S.D., BREWER, E.A., and AMIR, E. 1996, “Adapting to Network and Client Variability via On-Demand Dynamic Distillation”. *Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Opererating Systems (ASPLOS-VII),* Cambridge, MA 160-170.

FRANKLIN, C. 2005, “How Bluetooth Works”. Available from: http://electronics.howstuffworks.com/bluetooth.htm. [Accessed 2005].

FREIRE, J., KUMAR, B., and LIEUWEN, D. 2001, "WebViews: Accessing Personalized Web Content and Services", *In Proceedings of World Wide Web 10,* pp. 576-586.

GAST, M. 2002, “802.11 Wireless Networks: The Definitive Guide”. O'Reilly, 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA. ISBN: 0596100523.

GUPTA, S. 2005, “Context-Based Content Extraction of HTML Documents”*,* Doctor of Philosophy, Columbia University.

GUPTA, S., KAISER, G., NEISTADT, D., and GRIMM, P. 2003, "DOM-based Content Extraction of HTML Documents", *Proceedings of WWW'03 Budapest, Hungary, May 20-24*.

GUPTA, S., KAISER, G., NEISTADT, D., and GRIMM, P. 2004, "Automating Content Extraction of HTML Documents", *World Wide Web Journal, January*.

HADJISOTIRIOUS, C., MARSHALL, K., and ANDREW, R. 2004, “ASP.NET Web Development with Macromedia Dreamweaver MX”. Apress Publishing, Berkeley, CA, USA. ISBN 1590593480.

HAN, R., BHAGWAT, P., LAMAIRE, R., MUMMERT, T., PERRET, V., and RUBAS, J. 1998, "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing", *IEEE Personal Communications*, vol. 5, no. 6.

HAN, R., PERRET, V., and NAGHSHINEH, M. 2000, "WebSplitter: a unified XML framework for multi-device collaborative Web browsing", *Proceedings of the 2000 ACM conference on Computer supported cooperative work* pp. 221-230.

HARADA, R. 2005, “Network Management for Wireless Data Collection Applications”. PSION Teklogix. Available from: http://www.psionteklogix.com/assets/downloadable/Network_Mgmt_for_WLANs.pdf . [Accessed 2005].

HARRISON, J. V. & ANDRUSIEWICZ, A. 2003, "The digital signage exchange: a virtual marketplace for out-of-home digital advertising", *ACM Conference on Electronic Commerce*.

HARRISON, J. V. & ANDRUSIEWICZ, A. 2004, "Using Wireless Networks to Enhance Narrowcasting in Public Spaces", *Proceedings of the Consumer Communications and Networking Conference 2004*.

INTEL 2003, “IEEE 802.16 and WiMAX - Broadband Wireless Access for Everyone”. Available from: http://www.intel.com/netcomms/events/downloads/ieee80216_wp.pdf . [Accessed 2004].

INTERNATIONAL ENGINEERING CONSORTIUM 2005, “Global System for Mobile Communication (GSM)”. Available from: http://www.iec.org/online/tutorials/gsm/ . [Accessed 2005].

INTERNET SECURITY SYSTEMS 2001, “Wireless LAN Security”. Internet Security Systems Technical White Paper. 6303 Barfield Road, Atlanta, GA 30328, USA. Available from: http://documents.iss.net/whitepapers/wireless_LAN_security.pdf . [Accessed 2004].

KUNZE, M. and ROSNER, D. 2001, An XML-based Approach for the Presentation and Exploitation of Extracted Information. *Proceedings of the First International Workshop on Web Document Analysis*, Seattle, WA, September 8.

LEWIS, B., and DAVIS, P. T. 2004, “Wireless Networks for Dummies”. Wiley Publishing, Inc. 111 River Street, Hoboken, NJ 07030-5774, USA. ISBN: 0-7645-7525-2.

LIN, H., CHEN, J., and JIANG, M. 2002, "Integration of GPRS and Wireless LANs with Multimedia Applications", *Proceedings of IEEE PacificRim Conference on Multimedia,* pp. 704-711.

LUM, W. Y. and LAU, F. C. M. 2002, "A Context-Aware Decision Engine for Content Adaptation", *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 41-49.

LYIJYNEN, M., KOSKINEN, T., LEHTONEN, S., & PESOLA, J. 2003, "Content Adaptation on LANE Active Network Platform", *Proceedings of the 7th International Conference Telecommunications (ConTEL)*, vol. 2, pp. 721-724.

MARKS, R. B. 2002, "Advancesin Wireless Networking Standards", *PacificTelecommunicationReview 4th Quarter* pp. 30-37.

MILROY, S., SAFFORD, D., COX, K., KALANI, A., & BARKER, L. 2002, ".NET Mobile Web Developer's Guide". Syngress Publishing, 360 Park Avenue South New York, NY 10010-1710 USA.

MINOLI, D. 2003, "Hotspot Networks - WiFi for Public Access Locations". McGraw-Hill, Two Penn Plaza, New York, NY 10121-2298. ISBN: 0-07-140978-5.

MOLTA, D. 2005, "Wireless Hotspots Heat up". Network Computing. Available from: http://img.cmpnet.com/nc/1409/graphics/1409f1_file.pdf . [Accessed 2004].

NEWHOUSE, T. and PASQUALE, J. 2005, "Java Active Extensions: Scalable Middleware for Performance-Isolated Remote Execution", *Elsevier Computer Communications Journal*, vol. 28, no. 14, pp. 1680-1691.

NORTEL NETWORKS 2003, "Introduction to Quality of Service (QoS)". Available from: http://www.nortelnetworks.com/products/02/bstk/switches/bps/collateral/56058.25_022403.pdf . [Accessed 2004].

OMNIVEX CORPORATION 2005, "What is Digital Signage?". Available from: http://www.omnivex.com/products/digital_signage.asp. [Accessed 2005].

OPIE, J. W. 2004, "Evaluating Existing Technologies for the Management of Digital Signage Networks". Computer Science Honours, Rhodes University.

PINHANEZ, C., KJELDSEN, R., TANG, L., LEVAS, A., PODLASECK, M., SUKAVIRIYA, N., and PINGALI, G. 2003, "Creating Touch-Screens Anywhere with Interactive Projected Displays", *Proceedings of ACM Multimedia,* pp. 460-461.

RAMAKRISHNAN, N. 2000, "PIPE: Web Personalization by Partial Evaluation", *IEEE Internet Computing*, vol. 42, no. 9, pp. 21-31.

RIDGEWAY, M. 2002, ".NET Wireless Programming". SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501, United States of America.

ROGERS, Y., PRICE, S., FITZPATRICK, G., FLECK, R., HARRIS, E., SMITH, H., RANDELL, C., MULLER, H., O'MALLEY, C., STANTON, D., THOMPSON, M and WEAL, M. 2004, "Ambient wood: designing new forms of digital augmentation for learning outdoors". *Proceedings of the 2004 conference on Interaction design and children: building a community table of contents*. Maryland, 3-10.

ROSCHELLE, J. 2003, "Keynote paper: Unlocking the learning value of wireless mobile devices". *Journal of Computer Assisted Learning*, 19, 260-272.

RUBINSZTEJN, H.K., ENDLER, M., and RODRIGUES, N. 2005, "A Framework for Building Customized Adaptation Proxies". *Proceedings of the IFIP conference on Intelligence in Communication Systems (INTELLCOMM).*

SENIOR CORPS TECH CENTRE 2005, "Benefits of the Rich Text Format". Available from: http://www.seniortechcenter.org/reference_library/software/benefits_of_rich_text.php. [Accessed 2005].

TANENBAUM, A. S. 2003, "Computer Networks". 4th edn, Prentice Hall, .ISBN: 0-13-066102-3.

W3C 2000, “Document Object Model (DOM) Level 2 Core Specification”. Edited by: Le Hors, A., Le Hégaret, P., Wood, L., Nicol, G., Robie, J., Champion, M., and Byrne, S. Available from: http://www.w3.org/TR/DOM-Level-2-Core/ . [Accessed 2005].

W3C 2004, “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies”. W3C Working Draft. Edited by: Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., and Tran, L. Available from: http://www.w3.org/TR/CCPP-struct-vocab/ . [Accessed 2005].

W3C 2005a, “Mobile Web Best Practice (MWBP) Working Group Charter”. Available from: http://www.w3.org/2005/01/BPWGCharter/Overview.html#Mission . [Accessed 2005].

W3C 2005b, “Scope of Mobile Web Best Practices”. Available from: http://www.w3.org/TR/mobile-bp-scope/ . [Accessed 2005].

WEB STANDARDS PROJECT 2005, “Introduction to Composite Capabilities / Preferences Profile (CC/PP)”. Available from: http://www.webstandards.org/learn/askw3c/feb2004.html . [Accessed 2005].

WIKIPEDIA 2005a, “Global System for Mobile Communications”. Available from: http://en.wikipedia.org/wiki/GSM . [Accessed 2005].

WIKIPEDIA 2005b, “Universal Mobile Telecommunications System”. Available from: http://en.wikipedia.org/wiki/UMTS . [Accessed 2005].

WEST, P., FOSTER, G., and CLAYTON, P. 2005, “Content Exposure of Slide Show Presentations for Selective Download and Annotation via Mobile Devices”. *Proceedings of the 4th International Conference on mlearning,* 25 to 28 October, Cape Town, South Africa.

ZURITA, G., & NUSSBAUM, M. 2004, "Computer-supported collaborative learning using wirelessly interconnected handheld computers". *Computers & Education*, vol. 42, pp. 289-314.