User Mediated Dynamic Online Adaptation for Mobile Devices

Walubengo Muliro Singoro

June 25, 2008
University of Joensuu
Department of Computer Science and Statistics
Master's thesis

Abstract

The amount are a lot of contents available on the internet today and the fact that these contents are mostly accessed using PCs is obvious. Increasingly though, there is need to access these content using *handheld mobile devices* (HMDs). This is necessitated by the fact that thus far the number of HMDs exceeds PCs compounded with the need to access the internet anytime and anywhere. Many challenges stand in the way of making contents available on the internet to HMDs. Firstly, the contents are designed and built for PCs which have superior features compared to HMDs. Secondly, HMDs come with different processing powers, screen sizes, screen resolutions, storage and network connections. This heterogeneity in HMDs might lead to problems if the same contents are sent to different HMDs. In order to overcome the challenges contents ought to be adapted to individual HMDs. At the center of it all user satisfaction needs to be taken into consideration. The strong argument being that the user is the end consumer of these contents and is also better placed to tell if the adapted contents are a fit.

In this study, we examine adaptation techniques in HMDs which we do by investigating the classifications given in the literature. By inference, further classifications are also suggested. The classification helps us group adaptors into different toxicological sets which we use later to single out few adaptors to analyze in details how they work. The analysis brings forth the weaknesses of existing adaptor. It is these weaknesses the design of UMDO (User Mediated Dynamic Online) adaptor strives to address.UMDO adaptor put forth in this study is to guarantee *Quality of Experience through* QoE user mediation which allows users to correct the adaptation when it too much or too little. UMDO adaptor is dynamic because it allows for adaptation rules to be built or corrected automatically and is online in the sense that these changes are applied instantly.

The ACM Computing Classification System:

H.5.4,H. 4.3,C.5,C.4

Keywords:

Handheld Mobile Devices, Quality of Experience, Adaptor

Acronyms and Abbreviations

ECCAM (Expert control content adaptation model) - a model proposed by Tong et al [45] for an adaptation engine to be used in education technology.

GPS (Global Positioning System) - – a satellite navigation system providing geographical position information that has global coverage.

IP Address (Internet Protocol Adress) - a number that uniquely identifies every independent computing device on the internet.

ITS (Intelligent Tutoring System) - a computer-based training system that customizes learning to each student based on the student´s abilities.

HMD (Handheld Mobile Device) - a gadget that is meant to be used for communication by users on the move while it is held in the palm.

PDA (Personal Digital Assistant) - a mobile device used to store and manage information (mostly personal information) to be accessed any time and any where.

PC (Personal Computer) - a computer designed to be used by one individual with no need of a computer operator whose role has mainly been taken over by the computer's operating system.

QoE (Quality of Experience) - level of the user´s subjective perception of the quality of services provided to him or her.

UMDO (User Mediated Dynamic Online) - an adaptor that adapts contents to given scenarios in real time and requires user intervention to perfect the adaptation process. Both current and past user interventions are used to seemlessly correct and improve adaptation.

VLE (Virtual Learning Environment) - software that is used as learning and teaching tool for learners and instructors respectively.

Contents

Chapter 1

Introduction

1.1 Overview

Content adaptation refers to the process of transforming contents so that they are suitable to a given scenario. The scenario in question can be a device or application presenting or playing the contents or network bandwidth through which the contents are transmitted. In this work we deal mainly with content adaptation to HMDs.

Millions and millions of HMDs such as mobile phones and PDAs are being produced and sold into the global market every year. There is no sign of this trend changing anytime soon. At the same time plenty of new information is being produced or generated for the internet every day. The contents currently on the internet are primarily designed for PCs which generally have wider screens with better resolution, more processing power, faster connections and wider and superior keyboards than HMDs like cell phones and *Personal Digital Assistants* (PDAs) [4]. This means that HMDs do not display or play some of the contents on the internet. It is also possible that because of too much adaptation, vital information is lost making the information played or displayed irrelevant.

There are more people with access to HMDs than there are those with access to computers [6]. Especially in developing countries there are more HMD users. According to a UN report, mobile phones are the only ICT where the number of users in developing countries has exceeded those in developed countries [46]. The reason why mobile phones have had higher penetration than computers in developing countries can be attributed to the nature of mobile phones. Mobile phones are cheaper, portable and have longer battery life which is necessary for places with poor electricity infrastructure like developing countries. Most internet contents that have been designed for computer having high processing power and big screens can not be played or displayed on HMDs. This means that people who have access to HMDs, who also happen to be the majority at the moment, can not access these contents. It is in light with this, that it is convincing to say that making contents on the internet available to HMDs will greatly increase content accessibility.

Making internet contents available available to HMDs presents a number of problems and challenges. First of all HMDs come with different features such as screen sizes and resolutions, processing power, network bandwidth capability and processing power. These features in turn determine what these devices can do and can not do. Firstly, this means that what one particular type and model of a HMD could display or play may not be displayed or played by other models or types. Secondly, different types and models of devices play different formats of media files. None of these HMDs can play or display all the formats of media type.

In line with this it will be important to come up with a way that will make contents currently inaccessible to HMDs, accessible to these devices. In so doing, it is not enough for the contents to just be displayed or played on the devices. The contents have to remain relevant to the user by not distorting vital information in those contents during transmission, manipulations or presentation. The contents also need to reach the user within a reasonable duration. At the same time the contents need to be as appealing to the user of a given device as possible relative to the original contents. Adaptation is one of the ways that have been put forward as an answer that will help to achieve these goals.

1.2 Motivation

Adaptation that is too much or too little is not good. Too much adaptation is bad because it might make the adapted contents which are a subset of the original contents incomprehensible or unappealing. This is would be because much of the original information would be left out after the adaptation process. Too little adaptation is equally bad, because presentation of adapted contents could be too slow. For instance, an image can take to long to download over a slow network hence affecting user experience negatively. Navigation or browsing through the contents could also be inconvenient due to adaptation. Adaptation of contents to a device will be useless if the users find the adopted contents incomprehensible or unappealing. This can be as a result of content adaptation process being slow or the gap between contents presented after manipulation and the origin contents being too wide.In other cases the system might just fail adapt the contents properly [19].

Adapting contents to many variable conditions leads to challenging problems in maintaining high levels of QoE leave alone quaranteeing it. QoE in this case is the level of the user´s subjective perception of the quality of content adaptation of contents provided to him or her after the adaptation process. Looking at existing systems and analyzing some of them in depth reveals that none of the systems has a way to guarantee QoE, is multiplatforms and can be deployed on large scale. Existing systems exhibit only one or two of the forementioned features but never all three. In this thesis I design a system, UMDO which aims to encompassed all three features. UMDO is an adaptor that adapts contents to given scenarios in real time and requires user intervention to perfect the adaptation process. Both current and past user interventions are used to seemlessly correct and improve adaptation. UMDO at best should guarantee QoE and at worst the system will point out to the content designers or system administrators that there are problems . UMDO will allow the user to get best presentation among the many options if they are any available in a timely manner. In addition, UMDO is a dynamic adaptation system because it is able to accommodated new scenarios such as a new device types or new technology as they arise without developers having to go back to the drawing board every now and then. This makes UMDO an effevective adaptation system that is multiplatform. UMDO is also designed in such a way that it can be deployed on large scale and in real-world beyond the laboratory tests.

The goal of this thesis is to present UMDO a system that goes beyond current research of personalization of contents to users and adaptation of contents to other conditions such as network bandwidth, device screen size and enviroment. UMDO's design incorporates not only personalization and adaptation but also aims to guarantee QoE.

1.3 Scope, Limitations and Constraints

This thesis is focused on contents adaptation in HMDs, the contents being any type contents that can be made to play or display on HMDS. In this case HMDs are all electronic devices that can be used while held in the palm for communication or computing purposes. Examples of such devices include cellphones, PDAs, smartphones and convergence devices.

In this thesis we will go through an analysis of existing classifications of adaptation techniques. Some new ways of classification are also proposed and best ways technique(s) put forward. This classification of adaptation techniques is limited to electronic contents being adapted to: the user, devices displaying or playing the contents, the network or location.

The main focus of this thesis is UMDO which is a proposed adaptation engine that is 'universal' in the sense that it to be used on any type of contents and for any type of HMD. The design of UMDO is independent from a particular transcoding algorithm, device or contents. It is because of this that algorithms for manipulation of contents into different variation are beyond the scope of this thesis.

1.4 Research Questions and Methodology

This thesis has three main research questions. The first question aims to establish what are the existing state of art techniques in content adaptation. Precisely put, the question is what are the techniques in content adaptation? In attempting to answer this question a literature review is required.

The second question is how do adaptors work? Literature review and in some cases actual testing of the adapters will help to answer this question.The third question is how can user mediation be used to enhance content adaptation? This question will best be answered by pointing out weaknesses in the current adapters and then putting forward ways of how to overcome these weaknesses.

In attempting to answer the aforementioned questions both formulative and evaluative approach will be used in the study. Formulative approach will come in when formulating the framework for user mediated dynamic online adaptation and evaluative approach is used in analyzing the envisaged performance.

1.5 Structure of the Thesis

This thesis has five chapters in total. Chapter one provides a preamble of what is expected in this thesis and why. In this chapter the thesis is introduced and for this an overview detailing a review of the thesis, motivation as to why this thesis is important and scope showing the limits of the thesis are given. The key research questions of this thesis and the methodology used in pursuit of the answers to the research questions are also introduced in chapter one. Chapter two presents the literature review in the field of adaptation techniques. This chapter is split in three parts; the first part defines what adaptation techniques are. The second part gives reasons why adaptation is necessary. The third part gives an account of background knowledge of different adaptation techniques based on the literature review. New techniques are also put forward and the best technique(s) recommented. Chapter three provides analysis of existing adaptors. In total five adaptors are analysed and discussed. UMDO is introduced then discussed in terms of the design and implementation in the last part of chapter three. In chapter four further discussion of UMDO adaptor is given and conclusion about UMDO system drawn. In discussing UMDO an analysis is provided in terms of expected performance and recommendation how the system should be built given. Overall conclusions and recommendations in this study are drawn in the last chapter, chapter five.

Chapter 2

Adaptation and Adaptation Technique in HMDs

2.1 What is Adaptation?

According to online Oxford English Dictionary [41] adaptation is defined as '*The action or process of adapting, fitting, or suiting one thing to another.*' In our case, the two main things we are concerned with when it come to adaptation are contents and HMDs. This being the case, we can then define adaptation as making contents suitable or fitting contents for use on HMDs. This definition is wide and vague. A more specific and technical definition of adaptation could be the ability of an application/contents to change the look, the feel or the behavior in relation to the context [26]. The context could be the user, the device in use, other applications interacting with the application in question, location, data, network bandwidth, physical enviroment, situation or time [34, 26, 15].

In defining adaptation it is important to define and differentiate between two words that could easily be confused. These words are adaptive and adaptable. Whereas by adaptive applications/contents we mean applications/contents being capable of automatically changing the look, the feel or the behavior in relation to the context, adaptable applications/contents are those that have the ability to allow another entity to change the application's/ contents' characteristics according to the entity's preferences. The entity in question could be the user, a device or another application.

2.2 Why Adaptation?

In this work, we center on adaptation techniques in portable mobile electronic devices. Why adaptation? See figure 2.1 which show a page from the net that is displayed on a HMD without adaptation, on the contrast figure 2.2 shows the same page on the same HMD only that this time adaptation is applied. Notice that the page in figure 2.2 is more readable. There are different reasons and views why adaptation is needed. Users of electronic devices such as PCs, PDAs and cell phones have different preferences and requirements hence the need for personalization of what is conveyed to them. The devices come with different features (see Table 2.2) which in turn affect their capabilities.

In addition, the interaction among these devices or with other apparatus is done through different types of communication platforms. For instance, communication could be wireless or wired. Each of these communication platforms will have different capabilities and options. What all these means is that there are many different scenarios that contents to HMDs could go through before they are presented to the user. Each of these scenarios, number of almost infinite, will require its own configuration and manipulation of contents if the contents are to be presented to the user in a time and acceptable manner. While it would be better to produce different versions at design time for different environments, the options are so many that it is impossible to exhaust all of them. In the event that it would be possible to produce

Figure 2.1: Simulation of a webpage displayed without adaptation on Opera Mini browser running on a HMD.

Figure 2.2: Simulation of a webpage displayed after adaptation was applied on Opera Mini browser running on a HMD.

Name	Display	Memory	Input method	Type
Apple iPhone 3G	LCD (Color TFT/TFD) Resolution: 320 x 480 pixels 3.5" diagonal	16 GB	Touch screen (operated using fingers)	Convergent device
Audiovox Thera / Toshiba 2032	LCD (Color TFT/TFD) Resolution: 240 x 320 pixels (QVGA)	32 MB	Touch screen (operated using stylus)	Smart phone
Nokia 1208	LCD (Color STN) Resolution: 96 x 68 pixels	4 MB	Keypad	Mobile phone
Sanyo Katana / SCP-6600	LCD (Color TFT/TFD) Resolution: 240 x 320 pixels (QVGA)	5 MB	Keypad	Mobile phone
Sony Ericsson K500	LCD (Color TFT/TFD) Resolution: 128 x 160 pixels	12 MB	Keypad	Mobile phone
Danger hiptop / T-Mobile Sidekick	LCD (Grayscale) Resolution: 240 x 160 pixels	16 MB	Touch screen (operated using stylus)	Convergent device
BlackBerry 8300	Type: LCD (Color TFT/TFD) Resolution: 240 × 260 pixels	64 MB	Keypad	Convergent device
Datalogic Falcon 4400	Active matrix TFT QVGA display; 320 x 240 pixels 3.5" diagonal	128 MB	Keypad	PDA

Table 2.2: Few examples of HMD and their key features that might influence adaptation

new contents manually for every existing scenario, there will be still new platforms and devices with different and new capabilities yet to be introduced to the market. At the same time it will not make economic or practical sense to produce contents meant for the masses for only one platform, user or device [12].

Contents come with different media and data types some of which would not be played on some of the HMDs without being adapted. In addition mobility of these devices means they will be used in different surroundings. These surroundings can be completely the opposite of one another. For instance the devices could be used in a noisy or quiet area, it could be a bright or dim environment, the user could be stationary or moving and so on. In some of these cases it will be necessary to adapt contents to suit the surroundings in order to maximize user satisfaction.

2.3 Classification of Adaptation Techniques

2.3.1 Adaptive hypermedia systems

Peter Brusilovsky provides an outline which shows the taxonomical view of adaptive hypermedia systems [8]. In this work, build on an earlier compehsive review by the same author, Brusilovsky breaks down adaptive hypermedia system into three distinct hierarchical classifications based on: area in which adaptive hypermedia is applied; what could be adapted and what adaptive hypermedia is adapted to.

Six areas of application in which adaptive hypermedia plays a role are put forward under the first category of classification by Brusilovsky. These areas are *educational hypermedia, on-line information systems, on-line help systems, information retrieval hypermedia, institutional hypermedia, and systems for managing person-*

alized views in information spaces. It is important to point out that since then other application areas, which could stand out on their own or are part of the six named application areas and were not there in 2001 or were not mainstream when Brusilovsky published his paper, have cropped up. One of such areas is online social networking websites such as MySpace, Hi5, Facebook and others. Social networking websites are adaptive in the sense that the sites customise and serve pages to the user based on the user's profile, friends and preferences. For instance, the sites could automatically recomment to a member new friends to add to friends list based on certain criteria. Also, the sites normaly restrict some areas on the site from being viewed by certain visitors to the site. Another application area of adaptative hyper media that is mainstreem now but was not mentioned by Brusilovsky is online advertisement. Advertisements on the internet are now days customised and served according to the internet user's profile, browsing history, physical location, time and so on. It is also likely that new areas where hypermedia can be applied will be identified and introduced in future.

The second category of classification of adaptive hypermedia system by Brusilovsky which is based on what could be adapted is broken down into two main groups, *adaptive presentation* and *adaptive navigation support.* Each of these groups is subdivided into subgroups which branch into further subgroups. The figure 2.3 captures this.

When it comes to what adaptive hypermedia is adapted to, the third category of classification put forward, Brusilovsky focuses only on the user. He suggests breaking this down into three main categories: *user data (user characteristics), usage data,* and *environment data* ([8] citing [25]). Besides the user, there are other things that adaptive hypermedia can be adapted to. Network bandwidth, appli-

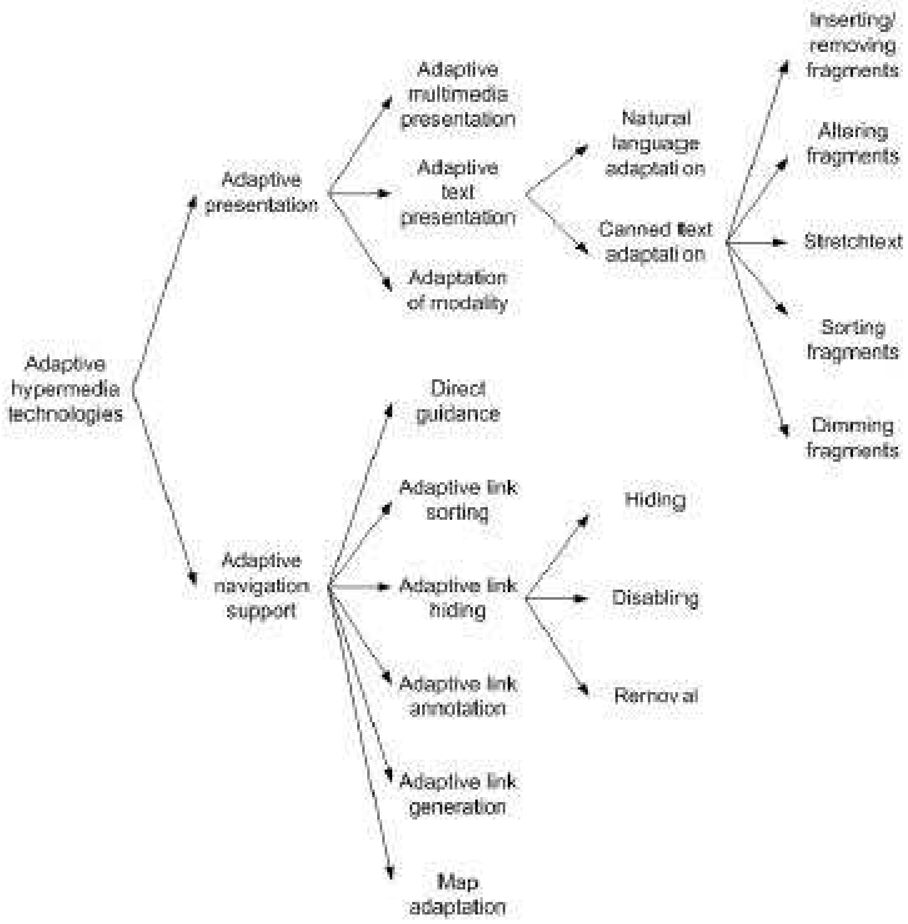

Figure 2.3: Brusilovsky's taxonomy of adaptive hypermedia technologies [8]

cations to manipulate the hypermedia and devices for playing and presenting the hypermedia to the user are some of the things adaptive hypermedia can be adapted to. Falling under device for playing and presenting hypermedia are HMD, television and personal computers. In fact, Brusilovsky in his work had put adaptation to HMDs under future trends. In our analysis of adaptation techniques we will focus on HMDs talking into consideration both adaptivity and adaptability. In this case adaptability (from adaptable) means users of the HMD being able to change the characteristic of what is presented to them according to their preferences ([24] cited in [29]). However, adaptivity (from adaptive) means the system changing automatically to accommodate the changing scenarios ([24] cited in [29]). Notice that Brusilovsky only talks about adaptivity and never about adaptability. Another striking difference between this work and Brusilovsky's is that in his classification Brusilovsky centers on hypermedia systems this work centers on contents, hypermedia being one of the subgroups.

2.3.2 Adaptation techniques in HMDs

Classifications of adaptation techniques in HMDs are varied and wide. Some of the classifications are explicit while others are implicit. Some of these classifications are clearly named and stated while others are inferred. These classifications are based on the following criteria: why, where, when or how adaptation is done and so on.

Adaptation in HMDs can be categorized as user adaptation or content adaptation. The difference between the two is based on why adaptation is done. User adaptation is personalization of contents to suit the user needs and preferences [8]. On the other hand content adaptation is modification of contents to suit the device displaying or playing them, the medium transmitting the contents or applications using the contents [19]. An example of user adaptation is adaptive hypermedia which is tailoring of navigation experience to match user preference based on facts such as user's previous navigation history, capabilities, knowledge, mood, surroundings or knowledge learned from other users. For instance a museum guide system can transmit information to the user based on the item s/he moving next to or a way from which is consider as a sign of interest or disinterest respectively [8]. In the same context suggestion for next artifact(s) to be visited by a user can be recommended based on previous artifacts visited by the same user. Another example of user adaptation is conversion of websites which are text oriented and meant for users with visual capabilities to audio oriented website for visually impaired users. An example of such a system is annotation-based transcoding system for nonvisual web acces [3] built by Asakawa, C. and Takagi, H.

Content adaptation can be done in response to only one aspect like client device or network bandwidth or to a combination of aspects. Systems advocating for different types of interfaces for different devices are designed to respond solely to client devices' parameters such as screen size or input capabilities. Examples of this are: a system that creates a tree structured representation of a website's outline for display on mobile devices [23] and a system that adapts web content for efficient browsing on PDAs [14]. These systems are designed to only respond to one type of aspect, device capability. A system could also be designed to respond to network bandwidth only. We can call such systems *aspect-specific systems*. On the other hand there are examples of systems designed to respond to multiple aspects when adapting contents, such as BARWAN [7]. BARWAN is built to adapt contents to network bandwidth and device in use. The user can also choose the quality of what is relayed to him or her. Such a system we could call a none-aspect specific.

He et al. classify content adaptation into two categories general purpose and content-type-specific [21]. Content type specific is when only one particular type of

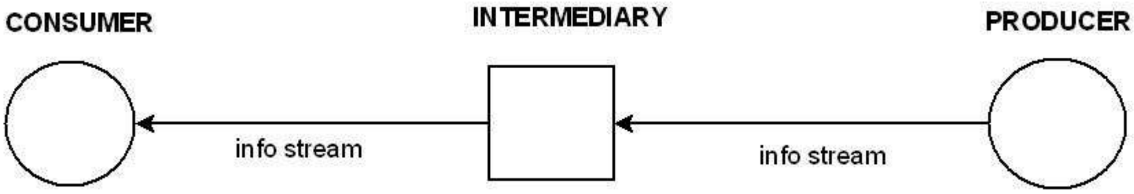

Figure 2.4: Intermediary Model [47]

content is the target of adaptation. The content types include images, video, text and sound. General type adaptation tends to encompass all the media types.

When it comes to where adaptation is done there are three areas this could happen: server side, client side or intermediary. Intermediary and proxy are used interchangeably.

Figure 2.4 shows *intermediary model* which depicts the server as producer, client as consumer, proxy as intermediary and info stream as content [47]. Each of these ways of implementation (server side, proxy and client side) has its own merits and demerits. The server side implementation allows the content producers to have control and the adapter in most cases is specifically tailored for the contents under consideration. The disadvantages with this type of implementation are that it is not scalable and it is only of use to contents residing on the host server. Proxy implementation is scalable and can be used on contents from different producers. The main disadvantage with proxy implementation is that content producers might not have control over the adaptation process. Meaning, they also do not have control on how the contents are presented to the user by the client and this might lead to copyright infringement(s) [33]. Client side implementation gives, as depicted in figure 2.5, the user and the device in use (the client) greater control unlike the other two types of implementations (server and proxy). Another advantage with the client side adaptation type of implementation is that there is no concern about the yet to be introduced client types or models. The downside of this type of implementation

Figure 2.5: Client side adaptation [39, Modified from]

is that client devices especially HMDs have limited capabilities hence limiting what they can do. In cases where they can do adaptation, processing speed and network bandwidth are brought into question. These two factors might affect user experience negatively. Having client-side adaptation also means that the software must be built to suit different platforms.

Classification of adaptation techniques could also be based on when adaptation is done. The two distinctions that could be thought of are: adaptation done at content design time and one done dynamically between the time contents are requested by the client and the time contents are presented to the user. Meawad and Stubbs [29] call the former off line content adaptation and the latter on line content adaptation. Style sheets can be used to provide many versions of the same contents which are forwarded to different types of clients, users, network bandwidths or any other scenario. This is an example of off line content adaptation. Dynamic adaptation is done on the fly and all content adaptations done at the client side or on a proxy server are dynamic.

Meawad and Stubbs imply that there are automatic adaptation techniques by saying that their adaptation system will be able to automatically update adaptation parameters. From this it can be inferred that they are adaptation techniques which are automatic or semi-automatic. Automatic is when during the adaptation process there is no human intervention. Semi-automatic on the other hand requires human intervention for the adaptation process to complete.

Adaptation techniques can also be divided along the lines of rule-based approach or constraint-based approach [35]. Rule-based approach as the name implies, consists solely of a set of rules that are built in the adaptation system. These rules are applied on the contents for adaptation to be accomplished. Constraint-based approach also consists of set of rules but in addition to that it also has options showing which rules are to be considered under given scenarios.

2.3.3 Best way to classify adaptors

We can notice that there is no classification technique that exists this far that will fully describes any adaptor. For instance an adaptor will always fall within two or more classification techniques. One illustration of this is BARWAN which is none-aspect specific, content adaptation that is both client side and proxy based adaptation engine. Hence calling it just by one classification leaves out other classifications.

Since all adaptors always fall between two or more techniques it would make sense then to combine existing classification techniques into tertiary techniques and describe adaptors according to such classification. We will then end up with names such proxy-based user adaptation, server side aspect specific adaptation, client side automatic content adaptation, server side rule-based approach adaptation and so on.

2.4 Adaptation Mechanisms

Adaptation in or for HMDs is done in different ways and depends on what is being adapted and for what purpose. Two main categories of what is normally adapted are content and navigation. Content adaptation mechanisms involve manipulation contents to be displayed and played to suit the users or the devices meant to display or play them. When it comes to navigation adaptation mechanisms, different or alternative navigation systems are given to enable users to navigate comfortably around the contents played or displayed to them on different devices. Between content and navigation adaptation mechanism the most researched on is content adaptation mechanisms. Content adaptation mechanisms are centered on adaptation mechanism of the four modalities of contents. Basically, the modalities are the four media types: text, image, video and sound. The adaptation of contents involves summarization or translation of these modalities. In this context, translation means converting the modalities from one modality to another, for instance, from image to text, video to image or video to sound. Summarization of modalities means; changing the structure, the shape or the size of the contents hence generating a new version of a given modality without translating into a different modality. Examples of summarization of modalities include text summarization, image compression, video compress and so on. Table 2.4 shows different modalities and how they are translated and summarized. It is important to point out that for each translation and summarization there exists different algorithms that are beyond the scope of this work. It is also important to note that under Content adaptation mechanisms, terms such as transcoding, resizing or slicing are normally used and fall within summarization, translation or both. Transcoding involves converting one format to another, transforming from one state to another or changing from one media type to another of media element(s) within the contents. For instance converting an image's format from JPEG to GIF, transforming a coloured image to a gray scale image or getting one picture frame from a video instead of the whole video. Resizing is the change of size of a media element. Resizing quite often involves making an image smaller before transmission or display. Slicing is implemented by cutting and leaving out (a) certain part(s) or element(s) of the original contents before transmission or presentation. An example of this is leaving out all the images within a web page when displaying it.

Navigation adaptation in MHD is necessary because contents on the internet for instance, are designed for laptops and desktop computers and not MHD which generally have small screen sizes, slow and inferior text input facilities, small storage capacities, limited battery life and slow CPU speeds. These facts have meant that pages have to be resized or reformatted to fit on the screens. Links, scrolling and text input reduced or eliminated to deal with slow or inferior text input facilities, small storage capacities, limited battery life or slow CPU speeds. To achieve these different mechanisms have been proposed or implemented.

When it comes to dealing with the challenges presented with small screen sizes, a number of approaches have been put forward. It has included main approaches such as compression of the whole page fit on the small screen then using the zoom-in and zoom-out option to view the contents with clarity [16], narrowing of the page to avoid horizontal scrolling and splitting of web pages into small related units that fit

Translation	Summarization
Image	
Translation	**Summarization**
Image to text	Size reduction
	Color depth reduction
	Resolution reduction
Video	
Translation	**Summarization**
Video to text	Spatial size reduction
Video to picture	Temporal size reduction
Video to sound	Bit rate reduction
	Play back rate reduction
Audio	
Translation	**Summarization**
Audio to text	Bit rate reduction
	Stereo to Mono
Text	
Translation	**Summarization**
Text to Audio	Key term extractions
Text to Text (Language translations)	Text summarization
	Document heading extractions

Table 2.4: Modality translation and Modality fidelity summarization [43, Adapted from]

on the screen then using links for access to each unit. Other approaches are use of vertical line as seen in Figure 2.6

Figure 2.6: Tree Control Shows Link Structure on a PDA [14]

instead of normal tree structures containing vertical lines or icons [14], use of lists instead of tables that generally need more horizontal space, folding and unfolding of contents that are grouped into units [13] and rotating of contents 90 degrees (also known as landscape mode) so that they can be viewed on a wider width. This is done because some MHD devices' screen heights are bigger than width so flipping the contents 90 degrees gives access to wider screens.

Slow and inferior text input facilities are compensated by extensive use of bookmarks, short cut keys and search engines to help locate web sites easily. In other cases widgets and aggregators are used to serve contents from different sites and pages at ago by summarizing and putting all the information on one page or in one frame. An example of such a system is I'mBored.com which is a prototype that was designed and built by Yahaya et al. [48]. It is also possible to avoid text input completely and in such a case, speech recognition is deployed.

Chapter 3

Design of UMDO Adaptor

User Mediated Dynamic Online (UDMO) adaptation is an on-demand (real-time) adaptation [7] system that allows user input to guide the adaptation process and at the same time seamlessly uses historical mediation data to perfect adaptation. User mediation means giving the users the ability to intervene whenever they feel the adapted contents have not been adapted well enough. In essence, it means allowing the user to ask the system to improve the adaptation whenever he or she feels further improvement is necessary. By allowing users to take part in adaptation process QoE is automatically monitored by the system. By monitoring QoE it is possible to maintain high QoE.

In designing UMDO adaptor it will be imperative to analyze existing adaptors so as to avoid reinventing the wheel. The selection of existing adaptors to analyze covers all the major categories of adaptation techniques in HMDs available. These techniques are dealt with in this work in the previous chapter.

3.1 Analysis of Existing Adaptors

There are many adaptation engines for HMDs today, both commercial [32, 22] and none-commercial [35, 18]. In this analysis I will deal with few but varied implementations. In order to get varied system we will look at system which implement different adaption techniques. In the analysis we look at lessons that could be learned from these implementations. The lessons will be important in designing a novel adaptable and adaptive system.

In this analysis we go through five adaptors. First is CDA which we analyze because of its similarity to UMDO. Both CDA and UMDO implement user involvement in the adaptation processes although the adaptation is done differently in each adaptor. Second, we look at Opera Mini browser. It is the largest known proxy based adaptation for HMDs. It is deployed commercially unlike other adaptors analyzed in this work or other similar work. Third to be analyzed is MobiGlam which is a large scale adaptation system used in education technology area. Like CDA it permits user involvement in order to improve adaptation. However, in CDA, user response is implicit while in MobiGLam it explicit. Also, in CDA, user input is used to improve or correct adaptation immediately while in MobiGlam they are only used later in redesigning the system. The first three adaptors are mainly content adaptors but the fourth adaptor to be analyzed is not. TenseITS falls under user adaptation. ECCAM the last to be analyzed is chosen because it is the first model in education technology to recommend an adaptation engine that allows any new user or HMD to be added to the system easily. This is possible because it does not require each individual user's preferences and device's capabilities to be stored anywhere.

3.1.1 Community Driven Adaptation (CDA)

Community Driven Adaptation (CDA) was proposed and implemented by Mohomed et al. in 2004 [35]. This novel application is a non-commercial proxy-based implementation that adapts contents as a result of implicit user feedback. It is mplicit because the feed back is not directly provided by the user but instead the system makes decisions based on history of how the users in the past have navigated through and altered the adaptation. With CDA it is assumed that users with similar characteristics such as device features, application being used and preferences can be grouped into groups called communities. It is further assumed that members of the same community would want to be served the same version of adapted contents. For a given community, CDA dynamically learns the most sought after version of adapted contents based on usage historical data stored in the databank. The version picked will be the one to be served to any given user in that community. In case of missing historical data in the databank the system falls back to rule-based or constraint-based system.

CDA evaluation involves three image-rich websites and the testing was done on laptops over a network with limited bandwidth. The bandwidth was the resource which adaptation was being done for and images were the media type being adapted. In general, 28 users that participate in the testing were presented with images with different clarity and allowed to refine each of them, by requesting new versions, to levels satisfactory to them. The nature of the tasks on the website required this to be done in order to complete the tasks. For instance on one website users were required to look at car pictures and then write down the number plate numbers. Since the initial pictures were of low fidelity this meant users would keep requesting for pictures of higher fidelity until the numbers were legible to them.

CDA system is said to provide 90% drop in network wastage and as high as 50% fall in user requests to improve image quality as rule-based or constraint-based systems. However, there exist shortfalls with CDA system. These shortfalls if corrected will provide a robust adaptation system.

The CDA system that was built and tested relied solely on implicit user response and static goal-driven contents that required the user to implicitly adjust the quality of images to satisfactory levels by implicitly requesting changes. Unfortunately this is not true reflection of reality, since most contents available for adaptation are not as goal-driven as the three websites built for CDA. In addition, not all contents are static, some are dynamic. This means if CDA was to be used in any large scale adaptation of contents different in nature from the three websites that were used it solely rely on implicit user response. The problem is that implicit user response on its own can be ambiguous [35]. This ambiguity means it would be hard (even impossible) for the system to learn users' preferences. Learning user preferences is core to CDA and without this capability CDA will not be of any advantage.

CDA system that was put forward was aspect specific because the only aspect it was being adapted to was the network bandwidth. This makes CDA a non-universal adaptor because there are several other aspects that need to be considered in adaptation of contents especially on mobile hand held devices. These aspects are influence by heterogeneous nature of the devices used in accessing the contents. These devices have different screen sizes, processing power, storage space and can handle different formats/types of data [44]. Lack of CDA to deal with any of other aspect is exacerbated by use of solely laptops in test cases.

CDA system advocates for building of communities by suggesting that users should coalesced into groups based on their preferences. This might be practically impossible to achieve due to users constantly changing their preferences, users having overlapping preferences and dynamic systems. In addition some systems by nature are used very few times per users. For instance, museum visitors using a museum system might only use the system once. It would be harder grouping such users into communities based on adapted contents preferences. In line with this I recommend structuring of communities based on device capabilities. These communities are easier to built and maintain. Key reason being that devices have clearly defined and unchanging parameters that are measurable and quantifiable such as screen size/type.

3.1.2 Opera's browser for mobile devices

Opera Mini and Opera Mobile are browsers for HMDs [37]. Opera Mini is free and Opera Mobile is not. Both browsers are developed and supplied by a Norwegian company called Opera Software. The two browsers are not open source.

A number of features differentiate between the two browsers. Opera Mobile runs on smart phones and PDAs. On the other hand Opera Mini runs on all kinds of HMDs e.g. mobile phones, smart phones and PDAs as long as they are Java enabled. Opera Mini handles its adaptation on a proxy server [27, 38]; however Opera Mobile adapts everything at the client side (on the client device)[37].

Figure 3.1 illustrates this difference in adaptation techniques between Opera Mini and Opera Mobile. The difference can be explained by the fact that Opera Mobile is solely meant for devices that have higher processing power. For instance smart phones are relatively more powerful and have more capabilities than mobile phones.

Figure 3.1: Differences in adaptation between Opera Mini and Opera Mobile [39]

In this analysis we will focus on adaptation techniques in the latest version of Opera Mini 4 which is free and hence easier to access. In addition Opera Mini has a wider reach because it can be used on more devices than Opera Mobile.

Opera Mini browser works by requesting web contents through proxies hosted by Opera Software Company or in some cases mobile operators. The contents which are served by content servers are sent to the Opera Mini browser through the proxies. These proxies strip down the websites' contents to smaller sizes by transcoding, compressing and rendering pages before forwarding them to the Opera Mini browser. Among the things done by the proxies is resizing images, frames and columns of text on web pages to the HMD screen's width. This linearises the page contents while keeping background colours and images intact making scrolling horizontally unnecessary. This technology is called Small-Screen Rendering (SSR) [40, 44, 5].

On top of content adaptation Opera Mini minimally implements user adaptation as well. This is done by synchronizing user's activities on the internet that are done using the HMD with those done by the same user using a PC. For instance the bookmarks kept by Opera Mini on a HMD can automatically be synchronized to be the same at all times as those kept on the Opera Mini user's PC [38].

Opera Mini provides options for the user to mediate adaptation by changing his or her preferences. For instance, the user of Opera Mini browser can switch between 'Mobile Site Viewer' and 'PC Site Viewer'. In 'Mobile Site Viewer' mode the web page is reformatted to fit inside the mobile screen's width. In 'PC Site Viewer' mode the web page, although compressed, looks the same way it would appear on a PC.

Users of Opera Mini can change the fidelity level of images displayed on websites. One can set the quality to any of the three: ‘Low’, ‘Medium’, or ‘High’. Further, Opera Mini has provision for users to do away with images altogether. This they do by choosing images not to be loaded by Opera Mini. Figure 3.2 and Figure 3.3 show a website rendered using SSR technology. ‘Show images’ option is off for figure 3.2 and on for Figure 3.3. Figure 3.4 shows the same page as displayed on PC monitor.

User mediation can further be noted in the ability of the user to zoom in and out of images and text. Related to this is the ability of the user to fold and unfold a section within a page. When a section is folded, contents in that section are hidden and when unfolded the contents are exposed.

Another feature that users have control over and we can consider it to be client side adaptation is the ability for the users to flip around the pages to Landscape mode. This is done in cases where the width of the screen on the HMD is narrower than the height hence flipping the web page gives the user wider view of contents horizontally.

Despite many good features that come with Opera Mini 4, there are some notable shortcomings with this browser. Opera Mini 4 does not have capability to play videos on websites that have links to video files on them. The browser replaces videos with image clips. Another shortcoming that can be noted with Opera Mini 4 version is the user does not have sufficient explicit controls over the adaptation process. For instance the user can choose image quality of ’Low’, ’Medium’, or ’High’ only and nothing in between the three. A better approach would be to allow users to control image fidelity level the same way it is done in CDA [35]. Although

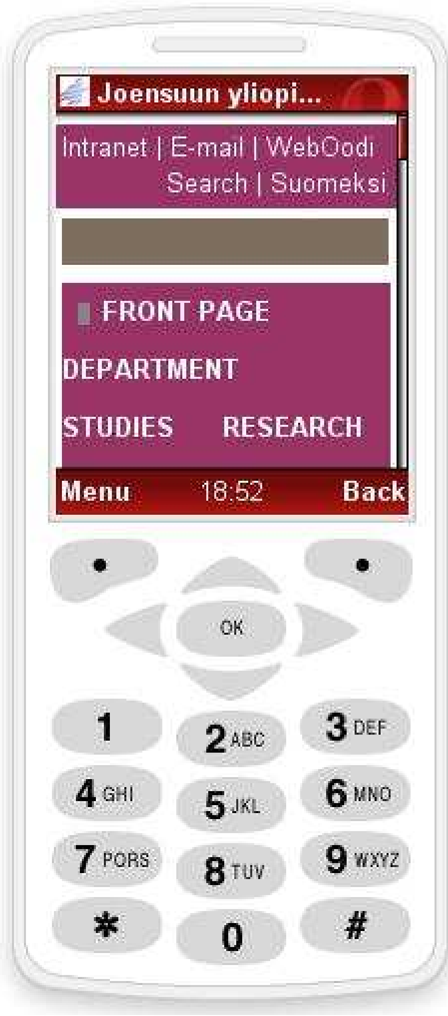

Figure 3.2: Web page reformatted to fit in a small with; Images are not displayed

Figure 3.3: Web page reformatted to fit in a small with; Images are displayed

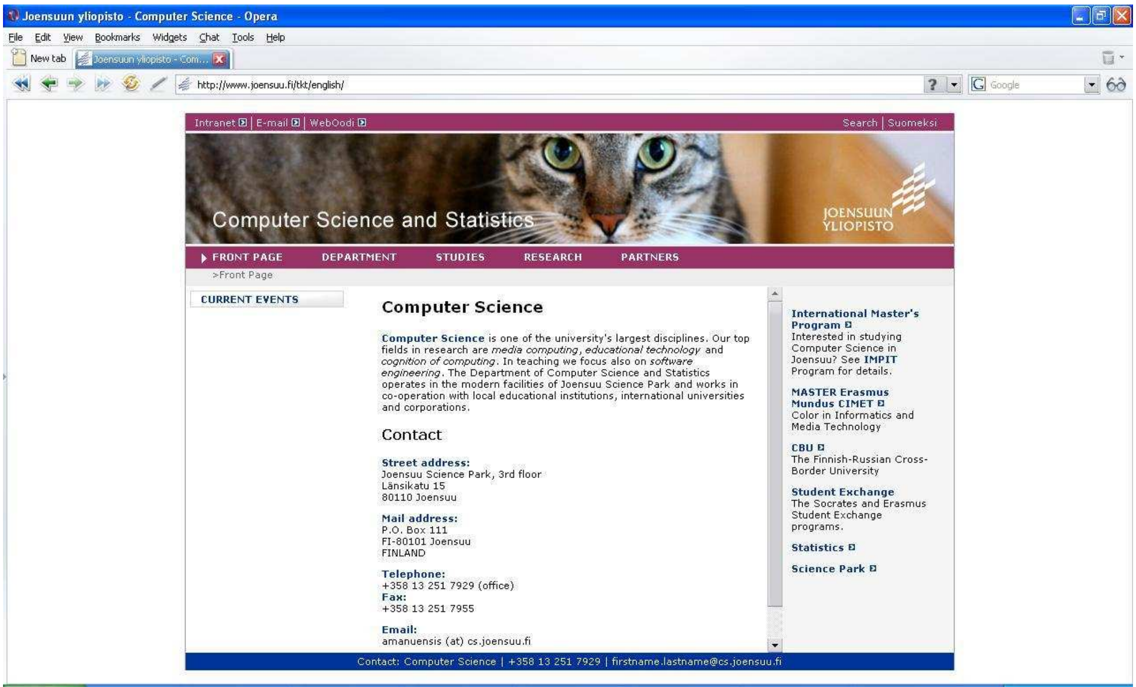

Figure 3.4: Web page dispayed on a PC monitor by Opera browser without adaptation taking place.

there is no way of telling since Opera Mini 4 is proprietary software, there is no evidence to show that users' preferences are learned by the system with the aim of improving the adaptation process. This can deduced from the fact that on installing the browser the default values for preferences remain at minimum unless changed by the user. It can be argued that if the system was learning from users' preference the default preference would be set to users most preferred settings. This argument is not conclusive though because it could be the case that minimum values for the settings are the most popular among the users. As a result of lack of explicit control on adaptation, the user cannot do anything if adaptation is not good enough. It seems this is also a shortcoming of Skyfire by Skyfire Labs [42] which is a close substitutes of Opera Mini. Skyfire is also proxy-based adaptation system. Other browsers for HMDs are Minimo by Mozilla Foundation [36], Safari for Apple iPhone [2], Deepfish by Microsoft Live Labs [30], Obigo by Teleca, Netfront by Access [1] and others.

3.1.3 MobiGlam

MobiGlam is a system developed at University of Glamorgan meant to extend the use of *Virtual Learning Environments* (VLEs) to HMDs. Though, MobiGlam is built to work with any type VLE so far documented evidence shows it has been only tested and used with Moodle. At the moment of writing this work MobiGlam is used at University of Glamorgan in the United Kingdom, Athabasca University in Canada and with work based learners in Wales in the United Kingdom. At the two institutions of higher learning and with the work based learners in Wales, MobiGlam is to used enhance learning through VLEs by bringing what would normally be done on computers to HMDs. Through MobiGlam users can interact and collaborate with each other, as well as update and be up to dated on courses offered through VLEs instantly anytime anywhere. Using MobiGlam user can access messages, forums, chats, lessons, workshops, assignments, grades, news, events, contacts, wikis, tips, quizzes etc. Figure 3.5 shows a simulation of MobiGlam displaying some of the features that can be accessed. [31, 28]

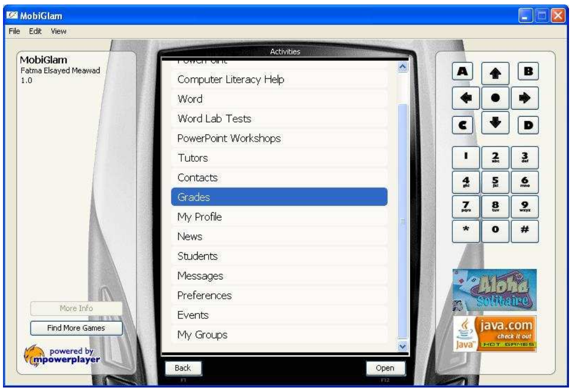

Figure 3.5: Demo of MobiGlam

Using MobiGlam users can access all features of a VLE that are supported by HMD in use. Personalization of features to HMD is based on device capability and user perception. For instance when using a HMD, *display and upload pictures* feature is only enabled if the device has those capabilities.

Technically, MobiGlam is an adaptive and adaptable system that adapts VLE to HMDs. This is achieved through use of a light-weight J2ME application installed on the client (HMD) and server side implementation. Browsing, editing and customization functionalities are provided by J2ME application while interoperation and adaptation of VLE is done by the server side implementation. Adaptation is mainly achieved through a decision engine, relying on Bayesian techniques. The decision engine makes the decision on what is to be adapted and how based on 'offline Bayesian Learning' of historical data collected about user experience. At the initial stage of deployment, MobiGlam's adaptation was considered rudimentary and not fully developed. At this stage, MobiGlam was a prototype which was mainly used to collect data about changing values of the adaptation parameters like bandwidth, media display, learner activity, response time and so on. It is from this collected data that the adaptation was later, in an off-line process, fine tuned by 'teaching' the decision engine what the optimum values for each of the parameters were. On completion of the 'learning' stage a complete system with updated and fine tuned adaptation engine was put in place. [29]

One of the differences between MobiGlam and CDA that we can note is, in CDA learning is online while that in MobiGlam is offline. One weakness with offline learning is that once the learning process is over, any new value or parameter that comes after will not be catered for. With so many different new HMDs with different parameter introduced in the market every now and then, offline learning might not be the most efficient way.

3.1.4 TenseITS

Bull et al. presents four mobile *Intelligent Tutoring Systems* (ITSs) that adapt education materials to learner's needs based mainly on the learner's knowledge, preferences and misconceptions [9]. In addition, in some cases location and synchronization between PC and HMD are also taken into consideration [9]. All four systems employ mainly client side adaptation with the main focus being the user [28]. The systems are TenseITS, C-POLMILE, MoreMaths and SQL-ITS. In this work will we will look at TenseITS because it seems to be the most written about in scientific literature [28, 9, 10, 17]. This is likely to mean that more information about it is available compared to the other three ITS.

TenseITS, was designed to be used for individualized learning by non-English speakers to help them learn the correct use of tenses while on the move or just away from the PC [9, 17]. It is built to fit into user's daily routine [10]. Interaction between TenseIT and the user is customized in accordance with user's location, their knowledge level, time available for learning and user's surroundings. Aspects being dictated to are either explicitly input to the system by the user or are worked out in the background by the system based on available data. Time available to the TenseITS user to use the system, the user's location and his/her surrounding (captured by user's ability to concentrate and the likelihood to be interrupted) are explicitly keyed into the system at the initial stage of use. Figure 3.6 shows the form used to key in the necessary inputs at the start of TenseITS. However, unlike the other aspects, the user's level of knowledge is automatically worked out by the system based on historical data that is collected during system usage. The user's level of knowledge as modeled by the system is to comprise of English language know-how, misconceptions and other hindrances associated with learning English as a second language.

TenseITS is built to presents mobile learners with four different types of individualized learning activities tutorial, revision, questions and feedback [10]. Figure 3.7 shows a feedback send to the user meant to address an underlying misconception. Each of the activities (tutorial, revision, questions or feedback) or a combination of activities is presented to the learner depending on their knowledge, location (surroundings) and time available for learning. For instance the system will offer a new tutorial to the learner if the learner has mastered the previous lessons, there is enough time for to learn a new tutorial, low chances of interruption and high probability for the learner to concentrate. However, if the time is short and chances of interruption are high then revision questions just enough for the time available are offered instead.

Main problem with TenseITS is that does not take into consideration adaptation to

Figure 3.6: Features for selecting location and location's contextual in TenseITS [17]

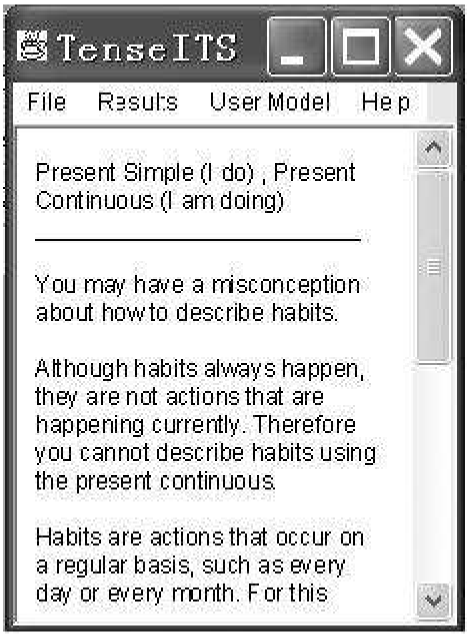

Figure 3.7: Excerpt from a feedback in TenseITS [10]

HMD hence it is likely that it can only be deployed on one platform, Compaq iPAQ Pocket PC. We can also note that TenseITS does not implement any mechanism to check if the user is actually satisfied with the way TenseITS adaptation works. In other words there is no way of to guarantying the QoE for an individual user. Other setbacks are those associated with client side based adaptation mentioned before in this work.

Further improvement on TenseITS could be done to include automatic location awareness by the system using GPS (Global Positioning System) positioning, and finding away to synchronize the system with the mobile learner's calendar. This should be done despite the fact that TenseITS is designed for the short time intervals between other activities and such time intervals are normally not keyed into the calendar [10]. It will also be better if TenseITS could be implemented on a PC and the two versions (PC and HMD version) made to work in synchrony for the each learner who might want to use both like it is the case with C-POLMILE [11].

3.1.5 ECCAM

Expert control content adaptation model (ECCAM) is the work of Tong et al. put forward in [45]. According to the authors of [45], ECCAM was a novel idea at the time when paper was published. Key to ECCAM's novelty is the fact that it is the first model to recommend an adaptation engine to be used in education technology that does not require each individual user's preferences and device's capabilities to be stored anywhere. Also, in this model user's preferences and device's capabilities do not have to be matched accurately for the adaptation process to take place. In fact, the model advocates for grouping of user preferences and device capabilities into clusters for which the adaptation engine uses to make its decision. For instance, any device with screen resolution less than 240*320 is assumed to be PDA and classified into a PDA class. Users are classified into three groups according to their education levels. Specifically, the three groups are Class 1, Class 2 and Class 3 which elementary students, primary students and advanced students, respectively.

ECCAM is a server side based adaptation that has an adaptation engine which makes its decisions and forwards contents based on the user and client device (HMD) classes. An example of this is if the client device is a PDA and the user is of elementary level then animation or video should be forwarded otherwise something else e.g a picture or just text should be forwarded instead.

ECCAM is implied to guarantee QoS but this might not be the case because it still sends the same version of contents to users or devices in the same groupings. A case to note which has been pointed out by the authors of [45] is that there is no guarantee that of users in the same group having the same preferences. It has also been noted that ECCAM centers only on user and device preferences and never on other aspects that might affect adaptation such as network connection and location.

3.1.6 Summary of analysed adaptors

Table 3.2 shows the key differences and similarities of adaptors analysed in depth in this study.

Metric

1. **System Nature** is either real world or laboratory. Real world mean the system has been deployed on large scale and is in actual use while laboratory means the system is has not been deployed beyond research and experimentation in controlled conditions.

2. **User mediation** is user intervention to perfect that is done in order to correct or perfect the adapted contents.

3. **Guarantee of QoE** is an assurance that the quality of adapted contents will be to the user not so different from the original contents.

4. **Commercial** means that the company owning the adaptation system has commercialized the system and tries to earn a profit from it.

5. **Client Side Adaptation** means adaptation system runs on the client device i.e. on a HMD.

6. **Server Side Adaptation** means the adaptation system runs on the same that servers the contents to clients.

7. **Proxy based Adaptation** means that there is a server called a proxy between the main server serving the contents and the client. The main purpose of the proxy in this case will be to adapt contents before forwarding them to the client.

8. **Content adaptation** refers to transforming content to adapt to aspects such as device and network but the user.

9. **User adaptation** is personalization of contents to suit the user needs and preferences.

10. **Tested on heterogeneous clients** is an indication whether the adapted has been tested on different platforms or not.

11. **Media type handled** shows the type of media(text, video, sound and image) the adaptor is built to adapt.

Metric	***CDA***	***Opera Mini***	***MobiGlam***	***TenseITS***	***ECCAM***
1	Laboratory	Real World	Real World	Real World	Laboratory
2	yes	minimal	offline mediation	no	no
3	minimal	no	no	no	no
4	no	yes	no	no	no
5	no	minimal	minimal	yes	no
6	no	no	no	no	yes
7	yes	yes	yes	no	no
8	yes	yes	yes	no	yes
9	yes	yes	yes	yes	yes
10	no	yes	yes	no	yes
11	images only	all except video	all	text only	all

Table 3.2: Summary of key metics of adaptors that we analyzed in details.

3.2 Realization UMDO Adaptor

In order to fully understand what UDMO entails we will first look at the problem description which will bring forward the reason why UDMO is needed. Next under architecture we will look at the system composition and design. Here we will break up the system into modules then show how these modules interact with each other. Lastly I will give recommendations on how and which platform UDMO should be implemented and this will fall under Implementation.

3.2.1 Problem description

Heterogeneous nature of HMDs means capabilities of these devices differ as well. Different capabilities mean different nature of contents to be handled and presented by these devices. It is also known that that these devices are used in different contexts which require variations in how the contents are presented to the user. Above all these, the user whom the contents are presented to needs to be satisfied and the goal why the contents are presented is supposed to be achieved. Content adaptation is recommended as a way to deal with heterogeneity of handheld devices and the different contexts they are used in. Content adaptation is further necessitated by the fact that most contents are designed for PCs which have more processing power, broadband connections and bigger screens than handheld mobile devices. It means contents have to be made 'small enough' to fit and be used on the HMDs. This act of making 'small enough' (adaptation) means only a subset of what was to be presented is presented [20] or (and) is presented in a way that it was not initially intentioned. This means the QoE will vary. The best way to monitor and improve QoE is by monitoring user satisfaction.

When it comes to user satisfaction it will be important to put in place ways of monitoring user perception on the adopted contents presented to him or her over the HMD. However, it is not sufficient for the system to just monitor users' perception of what has been adapted. A way to fine tune adaptation in case the users are not satisfied with what is conveyed to them should be put in place. This fine tuning should be as seamless and in real-time as possible. User mediation can be used to achieve this. In other words the system should allow the users to intervene to perfect adaptation. It will further help if the system had a way of keeping history of mediation attempts. This history can be used by the adaptation system to further streamline current and future adaptation by learning what the optimum values are for given devices and for predicting user preferences. Keeping history of mediation attempts and using it will also in the long run minimize user intervention.

In order to achieve this, an adaptation system that does on-demand adaptation, allows user intervention to guide adaptation whenever necessary, keeps track of user intervention (mediation) and uses information from previous user interventions to seamlessly streamline adaptation in real-time is needed. UMDO adaptor aims to achieve all these. UMDO is based on the reasoning that the users are in the best position compared to the adaptation system to establish and verify that contents are appropriately adapted and presented [35].

Another feature of UMDO, which is rare among the adaptation engines put forward so far, is that UMDO is intended for deployment on large scale and in all sectors (education, entertainment, medical, general, e-commerce and so on). This is achieved through the design that is sector and scalable.

3.2.2 Architecture

Since UMDO is meant to be implemented on large scale and is to be used in any sector, scalability is an important issue. From the literature review we saw that intermediary model of adaptation (proxy based) is the most scalable. Therefore, UMDO should be mainly proxy based. It is worthy noting that, to cover all cases UMDO should be adaptable and adaptive to all aspects adaptation in HMDs. The aspects are users, network bandwidths and types, client devices, applications and in some cases location and time. This means UMDO implements both user and content adaptation techniques.

UMDO will have two main distinct components. One component (UMDO client) will be on the client and another (UMDO server) on the proxy server. UMDO client is a thin client that aids the whole UMDO system to be adaptable. UMDO client presents the contents to the user. It also gives the user control over what and how the contents are presented to him/her by allowing him/her to manipulate the contents whenever necessary. Contrary to the thin UMDO client, UMDO server is a fat server housing the main components of the UMDO system. UMDO server is the adaptive part of the UMDO system

3.2.2.1 UMDO client

UMDO client provides the needed interface for the user to access the contents presented. It acts as a bridge the user and UMDO server. In so doing, it provides mechanism to the user for correcting the adaptation decisions made by the UMDO server. To be effective, the UMDO client has to be in constant communication with the UMDO server. UMDO client will communicate requests for contents, client device identification information, corrections decisions and any other needed information to the UMDO server.

3.2.2.2 UMDO server

UMDO server is the key part of the UMDO system where actually adaptation takes place. We can categorize UMDO server into seven parts *negotiation engine, decision engine, profiles database, content manipulation engine, adaptation smoothening engine, adaptation rules database* and *user interventions database.*

Negotiation engine interacts with the UMDO client, profile database and content servers like web servers. It is tasked with identifying the client device, composition of contents requested, network type, network bandwidth other factors that contents need to be adapted to. After identification, the negotiation engine then consults the profiles database to get values needed for adaptation on these factors. Lastly,

Category (Entity)	Parameter (Attributes)	Values
Device	Screen Resolution	Ranges of Resolutions
	Processor Speed	Ranges of speeds
	Input Choice	Keyboard, pen-based, keypad, voice
	Accepted data type	image , sound, video, text
	Accepted format	format options
	Memory	Ranges
	Operating System	Symbian, Windows CE, Linux etc
	Device Type	Phone, PDA, SmartPhone Network
Network	Communication Technology	MMS,SMS, GPRS, WLAN, Bluetooth
	Bandwidth	Ranges
	Response	Time ranges
	....	....

Table 3.4: Some entities and some attributes in the profiles database [29, Adapted from]

all the information is passed to decision engine for further action.

Profiles database should keep necessary profile information on facts that contents need to be adapted to. The information is needed to make adaptation decisions. Table 3.4 shows some of the entities and attributes that could be included in the profiles database.

The decision engine validates a request made by the UMDO client for contents. The aim is to ensure that the request is correct and is creditable enough to go ahead with the decision. The decision engine is also charged with the responsibility of deciding if adaptation on contents should be done and if so how and on what. This decision is based on information from negotiation engine and set of rules for the adaptation stored in the adaptation rules database. The set of rules in this case, state what should be done according to the information gotten from the negotiation engine. An example of such a rule could be if convert videos to images when the screen size is less than X. With information from the negotiation engine and set of rules from the adaptation rules database, the decision engine recommends how adaptation should be done to the content manipulation engine or sends the contents straight to the client. Contents are sent straight to the client if the decision engine sees that adaptation is not necessary.

Content manipulation engine performs content manipulation to come up with adapted contents which are then forwarded to the client. Content manipulation should be based on InfoPyramid framework that advocates for grouping contents according to their media types (video, text, picture and sound) known in this case as modalities [43]. During the adaptation process each of these modalities can be translated or summarized.

Adaptation smoothening engine only comes in after contents are initially presented to the user and the user is not satisfied with what is presented to him or her. At this point the user attempts to modify the adapted contents through UMDO client. The UMDO client then passes this information to adaptation smoothening engine which in turn generates information needed by the content manipulation engine to perform adaptation afresh. At the same time, adaptation smoothening engine updates user interventions database which inturn updates adaptation rules database. User interventions database is updated with records of the explicit choices the user made when attempting to control adaptation by trying to modify the adopted contents. On the other hand, adaptation rules that are stored in the adaptation rules database will be updated to reflect popular choices.

See Figure 3.8 showing an overview of the functional flow of information within UMDO system. The best protocol to be used for communication within UMDO system is the TCP/IP protocol because of two reasons. One, most servers with contents to be adapted communicate through TCP/IP protocol and two, TCP/IP protocol is connection-oriented. Being connection-oriented makes it easy and possible to monitor and synchronize communication between different components.

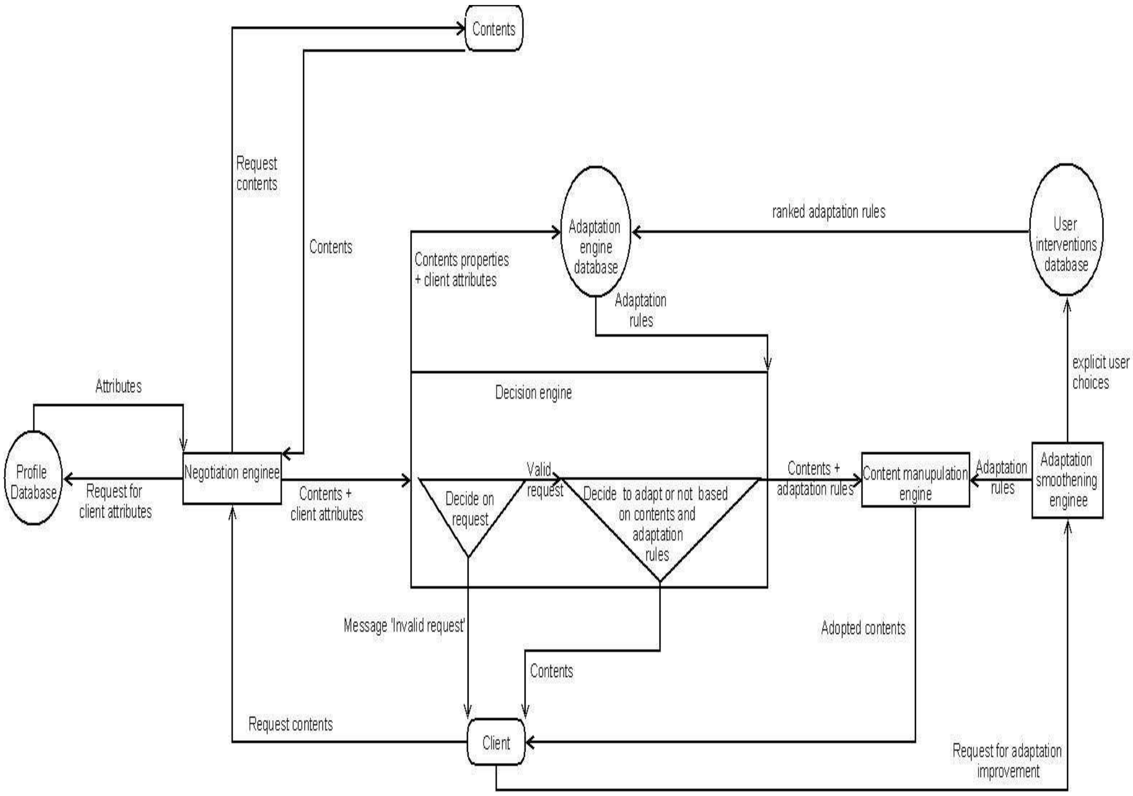

Figure 3.8: Functional achitecture of UMDO

3.2.3 How UMDO works

UMDO adaptation system involves three main entities: the user, the content server and UMDO system itself. The user's explicit roles as seen by UMDO are only two. The roles which are accomplished through the client device (HMD) are requesting of contents and guiding the adaptation process to do better whenever necessary. Hidden behind the guiding role the adaptation process is the user's role of assessing the adaptation process. It is this role that can be exploited to guarantee QoE. It should be noted that it is only through assessment that the user will be able to guide the adaptation system by requesting for better presentation of contents whenever he or she feels contents send to are not well displayed or played.

The content server which is on the other side of the UMDO system solo role is to provide the user through the client (in our case HMD) with contents needed. Although the server can server the contents to the user through the client as long as the user play his or her requesting for contents role, failing of contents to play or display correctly is possible. This is where UMDO comes in to check that request from the user is correct and contents are compatible with the client (HMD). If the contents served by the server are not compatible with the client the UMDO system adapts the contents to suit the client otherwise the contents are sent to the client as they are without adaptation. In some cases the user might not be satisfied with the contents sent over to the client and at this point the user is allowed to make suggestion as to how adaptation should be carried out. These suggestions are not only implemented immediately but are also kept for future use. UMDO ranks these suggestions based on their frequency and ranking in turn will be used to change adaptation rules. For instance, if users HMD with screen resolution 100 by 100 pixel request for images to be displayed in black and white (grey scale) more often then any future request by these kind of users will be served with black and white images. The option to change will still be there and it will be monitored as well. The users' suggestions history can also be used by content designers to improve on their designs. For instance, designers could redesign contents that have attracted many and frequent user interventions over a given limit over time.

Giving the users the option to control the adaptation will help maintain high QoE and by using data for future adaptation and content design decisions, QoE is guaranteed to be the best the user could get out of any system with adaptation mechanisms.

Chapter 4

Discussion of UMDO Adaptor

Here we go through analysis of UMDO as we look at expected performances deduced from documented analysis on performance of a similar system. Further improvements of UMDO are also put forward.

4.1 Analysis

The closest system that can be compared to UMDO is CDA. In fact UMDO can be seen as an extension of CDA both vertically and horizontally. Vertical extension is realized through UMDO introducing new functionalities that CDA does not have while horizontal extension is achieved by UMDO improving on some of the existing functionalities of CDA such as users being able to provide feedback on other modalities apart from just images. Among the major all new functionalities of UMDO that CDA does not have are the negotiation engine and the use of explicit feedback from the user. URICA [34] which is an improvement to CDA created by CDA creators does not have these functionalities.

Like CDA, UMDO is expected not to perform worse than other systems which are state-of-the-art because it is only an extension of such systems and it is meant to bring about further improvement in terms of performance. As we have noted, UMDO uses users' preference historical data and the immediate user's intervention to improve and perfect the adaptation process. This means that in the absence of users' preference historical data and lack of user intervention UMDO would perform just like the 'normal' content adaptor. In other words adaptation smoothening engine and user interventions database would not be taken into account.

There are notable challenges of UMDO that look on the surface as setbacks. First, user intervention is an extra burden to the user who will be required to do extra work other than what he or she is required to do. While this is true, it is also true with more users with the same kind of devices the system will be able to learn and perfect adaptation so that less or no intervention will be required at some point. Meaning, the more and longer UMDO the less user mediation is needed.

The second challenge is that there might be cases where user intervention might not work because of varied reasons. For instance, the rules in the adaptation engine might not be good enough or the contents might just not be adaptable. It should be pointed out that these issues can affect any adaptation engine. It should be further noted that unlike any other type of adaptor that does not support user mediation, UMDO can be easily made to detect such cases. This could be accomplished by monitoring and analyzing the data on user mediations in the adaptation process overtime. Monitoring and analyzing this data can give an indication of whether the adaptation is working or not. This can be as simple as monitoring the number of users attempting to intervene in the adaptation process against total number of users, frequency of mediation and number of mediations. How the data collected in such a process is analyzed is beyond the scope of this thesis. A point worthy mentioning here is that the monitoring mechanism can be made to alert producers of the contents and the designers of the adaptors whenever problems are detected or suspected.

The third challenge concerns legal issues, namely copyright and individual privacy. Contents that are to be adapted might be copyrighted by the content designers or content producers. In this case, it will be an issue altering the contents without consent from the designers or producers. W3C organization deals with the issue of copyright in content adaptation. When it comes to the issue of individual privacy, UMDO privacy policy should advocate for not collecting any personal data but use data that does not identify individuals. Identity of the client should be used instead, for instance IP addresses or MAC address.

UMDO advocates, unlike CDA, for grouping of users along devices' capabilities and this lessens the burden of trying to group users along other lines. The need for this is necessitated by the fact that it is not only easier to do this but this is already the norm. Designer already design contents for one group of users, PC users and hence adaptation can be used to reach out for other groups.

To improve the performance of UMDO in terms of speed caching of already adapted contents can be done on the UMDO proxy server so that any immidiate future request of the same contents by a client will not have to go through the full cycle of UMDO adaptation.

The nature of UMDO system calls for building it using the object oriented paradigm which advocates for using objects and interaction between the objects to design and built the system. UMDO system has clearly defined objects which can easly be defined as classes and implimented as such. Communication between these classes is inform of variables and value which past and shared among the classes.

Chapter 5

Conclusion and future work

The reasons to adapt contents that were not initially designed for handheld mobile devices to handheld mobile devices are many and varied. The urgency to have an efficient adaptor that is effective; adaptive; adaptable; deployable on large scale and can guarantee quality of service can not be over stated. In order to get such an adaptor it is important to first search through existing ones to see if there is one that meets the above mentioned conditions. While searching it is important to group similar solutions together. In this study this is done by classifying adaptation techniques. Classification of adaptation techniques is done according to why, where, when, how and on what adaptation is done. In addition, a better way that incorporates all the classification techniques recommended.

Classification of adaptation techniques is not sufficient to prove whether or not existing adaptors meet the conditions needed for an efficient adaptor. As such, in this study five adaptors are analyzed in details. The adaptors analyzed in depth are varied by type, creators, time when created and how they work. From this analysis we establish that there is no adaptor that can be deployed across board on large scale that can guarantee QoE.

Lack of an adaptor that is deployable across board on large scale and can guarantee QoE leads to a proposal of UMDO system. UMDO design puts across a flexible, adaptive and adaptable adaptor that can be deployed on large scale. On evaluating the design based on similar system we notice that UMDO can also guarantee QoE to the user by allowing him or her to intervene and correct the adaptation whenever necessary.

Future implementation of UMDO is necessary and some aspects could be investigated and expounded on in future to ascertain actual level of effectiveness of UMDO. For instance, factors that affect user's perception of QoE in UMDO could be investigated.

References

[1] ACCESS Co., Ltd. *Netfront browser*. `http://www.access-company.com/products/netfrontmobile/browser/index.html` (Accessed 06.03.2008).

[2] Apple Inc. *Safari browser for iphone*. `http://www.apple.com/iphone/features/#safari` (Accessed 06.03.2008).

[3] Asakawa, C., and Takagi, H. Annotation-based transcoding for nonvisual web access. In *Assets '00: Proceedings of the fourth international ACM conference on Assistive technologies* (New York, NY, USA, 2000), ACM, pp. 172–179.

[4] Bharadvaj, H., Joshi, A., and Auephanwiriyakul, S. An active transcoding proxy to support mobile web access. *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems* (1998).

[5] Blekas, A., Garofalakis, J., and Stefanis, V. Use of rss feeds for content adaptation in mobile web browsing. In *W4A: Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A)* (New York, NY, USA, 2006), ACM, pp. 79–85.

[6] Bracey, B., and Culver, T. *Harnessing the Potential of Ict for Education: A Multistakeholder Approach-proceedings from the Dublin Global Forum of the United Nations Information Communication and Technology Task Force (Ict*

Task Force Series). United Nations Information, United Nations Information, New York, USA., 2005.

[7] Brewer, E. A network architecture for heterogeneous mobile computing, 1998.

[8] Brusilovsky, P. Adaptive hypermedia. In *User Modeling and User-Adapted Interaction* (Netherlands., 2001), vol. 11, Kluwer Academic Publishers, pp. 87–110.

[9] Bull, S., Cui, Y., McEvoy, A., Reid, E., and Yang, W. Roles for mobile learner models. In *Proc. 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education* (2004), Y. Cui, Ed., pp. 124–128.

[10] Bull, S., Cui, Y., Robig, H., and Sharples, M. Adapting to different needs in different locations: Handheld computers in university education. In *WMTE '05: Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 48–52.

[11] Bull, S., and McEvoy, A. An intelligent learning environment with an open learner model for the desktop pc and pocket pc. In *International Conference on Artificial Intelligence in Education* (Amsterdam, 2003), IOS Press, pp. 389–391.

[12] Buttler, M. H. Current technologies for device independence. Tech. rep., HP Laboratories, Bristol, United Kingdom, 2001.

[13] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Seeing the whole in parts: text summarization for web browsing on handheld devices. In *WWW '01: Proceedings of the 10th international conference on World Wide Web* (New York, NY, USA, 2001), ACM, pp. 652–662.

[14] Buyukkokten, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. Power browser: efficient web browsing for pdas. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2000), ACM, pp. 430–437.

[15] Chen, G., and Kotz, D. A survey of context-aware mobile computing research. Tech. Rep. TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[16] Chen, Y., Ma, W.-Y., and Zhang, H.-J. Detecting web page structure for adaptive viewing on small form factor devices. In *WWW '03: Proceedings of the 12th international conference on World Wide Web* (New York, NY, USA, 2003), ACM Press, pp. 225–233.

[17] Cui, Y., and Bull, S. Context and learner modelling for the mobile foreign language learner. *Science Direct, System 33*, 2 (June 2005), 353–367.

[18] Foundation, A. S. Apache cocoon. `http://cocoon.apache.org/1285_1_1.html` (Accessed 19.02.2008).

[19] Freire, J., Kumar, B., and Lieuwen, D. F. Webviews: accessing personalized web content and services. In *World Wide Web* (2001), pp. 576–586.

[20] Fuschi, D. L., Parodi, E., Pearce, L., and Sutton, L. Content templating for automated content production and aggregation. In *Proc. Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution AXMEDIS '07* (2007), E. Parodi, Ed., pp. 231–238.

[21] He, J., Gao, T., Hao, W., and Yen, I.-L. A flexible content adaptation system using a rule-based approach. *IEEE Transactions on Knowledge and Data Engineering 19*, 1 (2007), 127–140. Member-Farokh Bastani.

[22] IBM. Websphere transcoding publisher. `http://www-306.ibm.com/software/pervasive/transcoding_publisher/` (accessed 19.02.2008).

[23] Jones, M., Buchanan, G., Thimbleby, H., and Marsden, G. User interfaces for mobile web devices www9 mobile workshop position paper. In *9th International World Wide Web Conference* (Amsterdam, The Netherlands., May 2000), vol. 9.

[24] Kinshuk, Patel, A., and Russell, D. Intelligent and adaptive systems. *Handbook on Information Technologies for Education and Training* (2001), 79–92.

[25] Kobsa, A., Koenemann, J., and Pohl, W. Personalised hypermedia presentation techniques for improving online customer relationships. *Knowl. Eng. Rev. 16*, 2 (2001), 111–155.

[26] Laakko, T., and Hiltunen, T. Adapting web content to mobile user agents. *IEEE Internet Computing 9* (2005), 46–53.

[27] Lehtonen, T., Benamar, S., Laamanen, V., Luoma, I., Ruotsalainen, O., Salonen, J., and MikkonenP, T. Towards user-friendly mobile browsing. In *AAA-IDEA '06: Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications* (New York, NY, USA, 2006), ACM, p. 6.

[28] Meawad, F. E., and Stubbs, G. A framework for interoperability with vles for large scale deployment of mobile learning. In *Proc. Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education WMUTE '06* (2006), G. Stubbs, Ed., pp. 13–17.

[29] Meawad, F. E., and Stubbs, G. Towards large scale deployment of adaptive mobile learning. In *Conference on Interactive Mobile and Computer Aided Leaning (IMCL2006)* (Amman, Jordan, April 2006).

[30] Microsoft Live Labs. *Deepfish browser.* http://labs.live.com/deepfish/ (Accessed 06.03.2008).

[31] MobiGlam. *MobiGlam Documentation.* http://mobi.etrainingwales.com/tikiwiki/tiki-index.php (Accessed 19.03.2008).

[32] Mobixell. Rich media service center. http://www.mobixell.com/content.asp?page=prod_rmsc (Accessed 19.02.2008).

[33] Mohan, R., Smith, J. R., and Li, C.-S. Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia 1*, 1 (1999), 104–114.

[34] Mohomed, I., Cai, J. C., and de Lara, E. Urica: Usage-aware interactive content adaptation for mobile devices. In *EuroSys '06: Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems 2006* (New York, NY, USA, 2006), ACM, pp. 345–358.

[35] Mohomed, I., Chin, A., Cai, J. C., and de Lara, E. Community-driven adaptation: Automatic content adaptation in pervasive environments. In *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 124–133.

[36] Mozilla Foundation. *Minimo.* http://www.mozilla.org/projects/minimo/ (Accessed 04.03.2008).

[37] Opera Software ASA. *Opera browsers.* http://www.opera.com/ (Accessed 02.03.2008).

[38] Opera Software ASA. *Opera Mini 4.* http://www.operamini.com/ (Accessed 04.03.2008).

[39] Opera Software ASA. *Opera Mini Images.* http://www.opera.com/press/images/mini/ (Accessed 04.03.2008).

[40] Opera Software ASA. *Opera's Small-Screen Rendering.* http://www.opera.com/products/mobile/smallscreen/ (Accessed 04.03.2008).

[41] Oxford University Press. *Online Oxford English Dictionary.* http://www.oed.com/ (Accessed 05.03.2008).

[42] Skyfire Labs. *Skyfire.* http://www.skyfire.com/ (Accessed 04.03.2008).

[43] Smith, J., Mohan, R., and Li, C.-S. Transcoding internet content for heterogeneous client devices. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS '98* (1998), R. Mohan, Ed., vol. 3, pp. 599–602 vol.3.

[44] TeliaSonera. Web content adaptation. Tech. rep., TeliaSonera Finland, Aug 2004. http://www.medialab.sonera.fi/workspace/WebContentAdaptationWP.pdf (Accessed 15.02.2008).

[45] Tong, M.-W., Yang, Z.-K., Liu, Q.-T., and Liu, X.-N. A novel content adaptation model under e-learning environment. In *Proc. th Annual Frontiers in Education Conference* (2006), Z.-K. Yang, Ed., pp. 1–5.

[46] UNCTAD. Ict indicators for development: Trends and impact. In *INFORMATION ECONOMY REPORT 2006. The development perspective* (New York and Geneva, 2006), U. NATIONS, Ed., vol. United Nations Conference on Trade and Development.

[47] Wang, H., Zhou, X., Zhang, Y., and Zhang, T. Information stream oriented content adaptation for pervasive computing. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 674–679.

[48] Yahaya, N. A., Gin, G. P., and Choon, C. W. Developing innovative web information systems through the use of web data extraction technology. 752–757.