Nelson Mandela Metropolitan University

for tomorrow

Department of Computer Science and Information Systems

Adaptive User Interfaces for Mobile Map-based Visualisation

Bradley Paul van Tonder

Supervisor: Prof. J.L. Wesson

December 2008

Submitted in partial fulfilment of the requirements for the degree of Magister Scientiae in the Faculty of Science at the Nelson Mandela Metropolitan University

Acknowledgements

I would like to thank my supervisor, Prof. Janet Wesson, for her support, advice and encouragement throughout this research. I wish to express my gratitude to her for the many hours she has spent reading the various versions of this dissertation and for the valuable feedback she provided. I would also like to thank the staff in the Department of Computer Science and Information Systems at NMMU for their valued feedback during the course of this research.

I would also like to thank Dr. Jacques Pietersen, from the NMMU Unit for Statistical Consultation, for his assistance in analysing and presenting the results of the evaluations conducted in the course of this research.

I would like to thank the NMMU Telkom Centre of Excellence for providing me with necessary funding and equipment to complete this research. I would also like to thank the National Research Foundation for their financial support.

Table of Contents

Summary

Mobile devices today frequently serve as platforms for the visualisation of map-based data. Despite the obvious advantages, mobile map-based visualisation (MMV) systems are often difficult to design and use. Limited screen space, resource constraints and awkward interaction mechanisms are among the many problems with which designers and users have to contend. Adaptive user interfaces (AUIs), which adapt to the individual user, represent a possible means of addressing the problems of MMV. Adaptive MMV systems are, however, generally designed in an ad-hoc fashion, making the benefits achieved difficult to replicate. In addition, existing models for adaptive MMV systems are either conceptual in nature or only address a subset of the possible input variables and adaptation effects.

The primary objective of this research was to develop and evaluate an adaptive MMV system using a model-based approach. The Proteus Model was proposed to support the design of MMV systems which adapt in terms of information, visualisation and user interface in response to the user's behaviour, tasks and context. The Proteus Model describes the architectural, interface, data and algorithm design of an adaptive MMV system.

A prototype adaptive MMV system, called MediaMaps, was designed and implemented based on the Proteus Model. MediaMaps allows users to capture, location-tag, organise and visualise multimedia on their mobile phones. Information adaptation is performed through the use of an algorithm to assist users in sorting media items into collections based on time and location. Visualisation adaptation is performed by adapting various parameters of the map-based visualisations according to user preferences. Interface adaptation is performed through the use of adaptive lists.

An international field study of MediaMaps was conducted in which participants were required to use MediaMaps on their personal mobile phones for a period of three weeks. The results of the field study showed that high levels of accuracy were achieved by both the information and interface adaptations. High levels of user satisfaction were reported, with participants rating all three forms of adaptation as highly useful.

The successful implementation of MediaMaps provides practical evidence that the model-based design of adaptive MMV systems is feasible. The positive results of the field study clearly show that the adaptations implemented were highly accurate and that participants found these adaptations to be useful, usable and easy to understand. This research thus provides empirical evidence that the use of AUIs can provide significant benefits for the visualisation of map-based information on mobile devices.

Keywords: Mobile map-based visualisation, adaptive user interfaces, model-based design, field study.

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Maps have been used for thousands of years to describe our physical environment. During this period, maps have also served as essential tools for planning and navigation. The advent of computers and the subsequent development of graphical output capabilities provided a new platform for the delivery of electronic maps to users. The first widespread example of this was the specialised Geographic Information Systems (GIS) which emerged in the 1970s and 80s (Reichenbacher 2004). These digital maps provided users with new capabilities, allowing them to search for features and landmarks, to visualise and analyse geographic information and to customise the appearance of maps. Map data could be stored independently of its presentation, allowing for easier updating and searching of maps. The evolution of the Internet has led to the development of simpler, mass-market GIS and a number of new applications have emerged for digital maps (Skupin 2000). Today, users can make use of web-based GIS to access driving directions[1], search for landmarks[1], visualise traffic volumes[2] and see collections of photos in a geographic context[3], to name but a few examples. Digital maps are, however, no longer only available on desktop computers.

A recent development is the provision of map-based applications to users of mobile devices (Chittaro 2006). Mobile phones have evolved from mobile equivalents of the traditional telephone to platforms for the delivery of information and services in a mobile context. A wide variety of Location-Based Services (LBS) are emerging, which aim to provide users with information or services relevant to their current position (Zipf 2002). LBS take advantage of a new generation of mobile devices which allow the user’s current position to be accurately determined using technologies such as Assisted GPS (A-GPS). Many LBS rely on maps as a core component.

A wide variety of map-based applications have been developed for mobile devices, many of which visualise information in a map-based context. These mobile map-based visualisation

---

[1] http://maps.google.com

[2] http://www.trafficmap.co.uk

[3] http://www.flickr.com

(MMV) systems visualise information by overlaying or integrating it into a map view of a geographical area. Some MMV systems have been developed to target the general consumer, while others are specialised applications for aiding workers in a particular profession. Consumer applications include simple map applications, applications which provide the user with navigation assistance and applications which help the user find points of interest (such as the nearest restaurant). Applications of this nature are also proving successful commercially. A survey in 2007 revealed that LBS, map and directions downloads together make up 40% of revenue earned from users downloading content onto their mobile devices in the US (Telaphia 2007). The same survey also revealed that users were willing to pay higher prices for location-based information than for downloads falling into other categories. Despite the obvious benefits, MMV systems are often difficult to design and use.

Designers and users of MMV systems are faced with several difficulties due to the nature and capabilities of mobile devices. Maps are large, while most mobile devices have displays no bigger than the palm of a human hand. Furthermore, mobile devices have limited memory and processing resources (Chittaro 2006). Mobile devices also provide the user with awkward means of interacting with the device (Al-bar and Wakeman 2001). Users of mobile devices are often distracted and divide their attention between their mobile device and their surroundings. These factors suggest that special measures need to be taken to ensure that MMV systems provide users with easy to use interfaces. Visualised information needs to be organised correctly and displayed in a format which matches the user's needs in order to make optimal use of the limited resources available. Adaptive user interfaces (AUIs) provide a possible means of tailoring MMV systems to each user's specific needs.

AUIs are designed to adapt to the characteristics of the individual user (Kühme 1993). As early as the 1980's, researchers realised the need to address the individual needs of each user, rather than designing generic, static interfaces (Murray 1987). Users have different capabilities, motives, goals, skills and interests, so the idea of providing users with personalised user interfaces holds obvious benefits.

The concept of a user model is fundamental to the creation of an AUI (Langley 1999). A user model contains knowledge which the system has acquired concerning the user. This knowledge can be obtained either explicitly (by asking the user to answer questions or complete test cases), or implicitly (by recording user interaction with the system). Implicitly acquired data is typically processed in order to gain useful knowledge about the user's interests and/or capabilities. This knowledge is stored in the user model and used to present a personalised interface to the user.

Adaptation has previously been recognised as a useful means of overcoming the limitations of mobile devices (Al-bar and Wakeman 2001). In the context of MMV systems, several types of adaptation provide potential benefits, including filtering information (Langley 1999) and adapting the presentation of visualisations according to user preferences (Koelle 2004). AUIs can potentially provide the means to overcome (or at least alleviate) some of the problems and shortcomings of MMV. Existing adaptive MMV systems are, however, generally designed in an informal fashion and the benefits achieved are often difficult to repeat. Model-based design can be used to address this problem.

The use of repeatable processes in the form of software development methodologies is often emphasised as an integral part of developing new software systems, reducing risk and ensuring that systems function as intended (Whitten *et al.* 2004). These methodologies advocate the use of models to formally define the context, behaviour and data of software systems (Sommerville 2006). The different aspects of a system are often communicated through the use of a modelling language (Gomaa and Hussein 2007) which provides standardised notations and diagram types. The Unified Modelling Language (UML) (Rumbaugh *et al.* 2004) has emerged as a popular modelling language in recent times (Van den Bergh and Coninx 2004). Models are often developed in the requirements definition and analysis phases and then used as the templates for implementation. Models can also be developed from existing systems, capturing good design practice and essential characteristics of a class of systems. Such models can then be used as a basis for the design of future systems (Sommerville 2006).

1.2 Situation of Concern

The limitations of mobile devices result in usability problems which negatively affect the perceived usefulness of MMV systems. Limitations in terms of screen size, hardware resources and interaction mechanisms, (Al-bar and Wakeman 2001; Karstens *et al.* 2003; Chittaro 2006) detract from the effectiveness, efficiency and user satisfaction of such systems. According to the Technology Acceptance Model (Davis 1989), perceived ease of use influences the perceived usefulness of a software system. This theory suggests that the usability problems of MMV systems can also impact on the perceived usefulness of such systems.

Existing research which addresses the above problem has followed two approaches. Firstly, existing visualisation techniques have been modified and new techniques have been developed to address the limitations of mobile devices. Techniques for the presentation and navigation of map-based information spaces have been developed which specifically target mobile devices (Harrie *et al.* 2002; Rosenbaum and Schumann 2005). These techniques focus on a particular aspect of the problems of MMV and do not consider the needs and preferences of the individual user. The second approach involves the use of AUIs in MMV systems.

Research into the use of AUIs in MMV systems can be categorised according to the variables which influence adaptation and the elements of the MMV system which are adapted. User characteristics and behaviour, context, tasks and the capabilities of mobile devices can all influence the adaptations which are performed (Mitrovic *et al.* 2007). Several aspects of MMV systems can also be adapted, including the information visualised, the visualisations and the user interface (Reichenbacher 2004). Potential benefits include more effective use of limited screen space, optimal use of limited processing power and memory capacity and reductions in the amount of interaction required from the user. Several adaptive MMV systems have been developed, but these systems have largely followed ad-hoc approaches, making the benefits achieved difficult to replicate.

Several models, frameworks and architectures for adaptive MMV systems have been presented. Reichenbacher (2003) presented a conceptual framework for adapting the information, visualisations and user interface of an MMV system according to the user's context. Models have also been presented for providing content recommendations and performing visualisation adaptation in mobile tourist guides (Cena *et al.* 2006). Research has also focused on adapting various aspects of the maps themselves, including a framework for adapting a wide variety of map display parameters to the user's preferences and context (Zipf 2002).

Existing models, frameworks and architectures have several limitations. Some are high-level conceptual frameworks, discussing adaptive MMV systems from a theoretical perspective, but lacking technical details regarding the implementation of such systems. Others are only suited to a very narrow class of systems. Existing models also only cover a sub-set of the adaptation input variables and effects to be found in existing research. A new model is needed which addresses these limitations.

1.3 Research Objectives

The primary aim of this research is to develop and evaluate an adaptive MMV system using a model-based approach. In order to do so, a model will be developed which facilitates the design of MMV systems which adapt in terms of information, visualisation and user interface according to the user's behaviour, tasks and context.

The following are objectives of this research:

- To identify the problems and shortcomings of existing MMV systems and techniques;
- To determine how AUIs can be used to enhance MMV systems;
- To develop a model for the design of adaptive MMV systems;
- To implement a proof-of-concept prototype based on the proposed model; and
- To determine the benefits provided by the prototype to users.

1.4 Scope

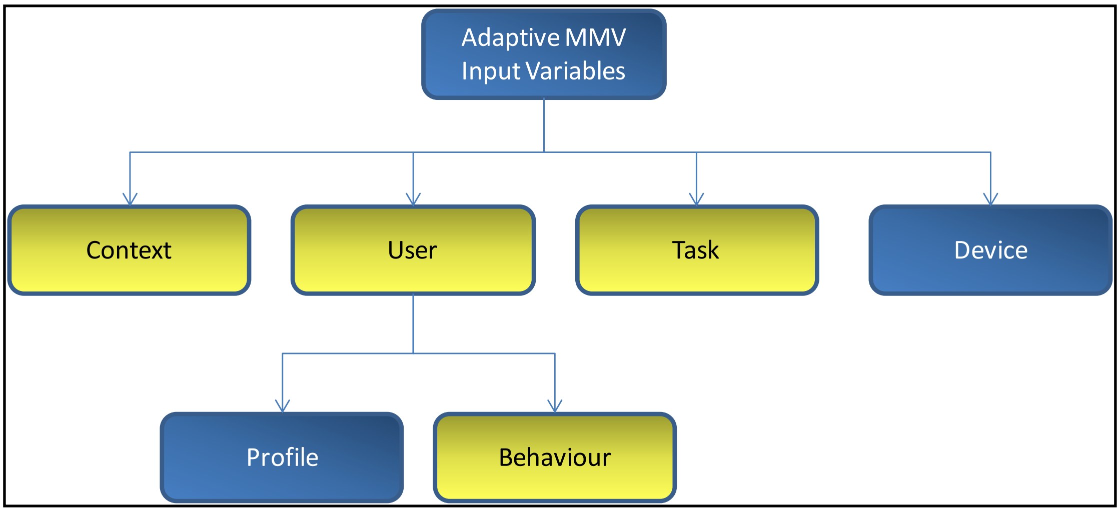

Figure 1.1: Input variables influencing the adaptation of MMV systems (variables within the scope of this research shown in yellow)

Existing research into the use of AUIs in MMV systems has focused on adapting to the capabilities of different devices (Reichenbacher 2004; Cena *et al.* 2006). This class of adaptation will not be considered in this research. Adaptations of this nature are largely technology dependent and therefore of limited value. The fast pace of change in the mobile phone marketplace and the wide range of devices, operating systems and interaction mechanisms available make it extremely difficult to develop a lasting solution to the problem of cross-device compatibility. The adaptation input variables to be considered in this research are shown in Figure 1.1. The users, their tasks and context will be considered as input variables. Adaptation to user behaviour, rather than static user profiles will be emphasised.

Adaptation of the information, visualisation and user interface will be the focus of this research, as shown in Figure 1.2.

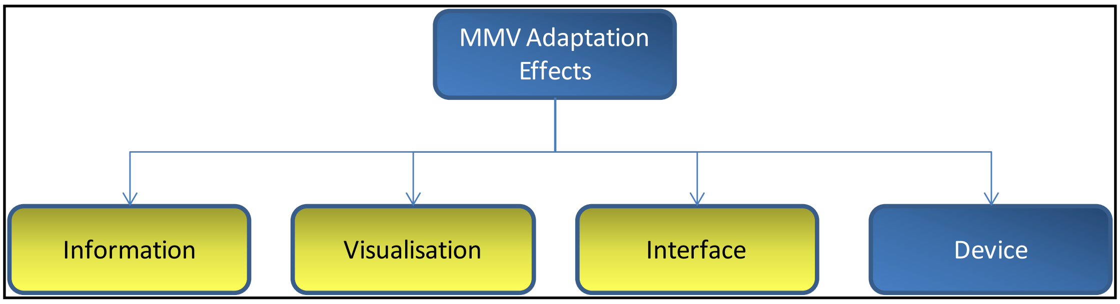

Figure 1.2: Adaptation effects (variables within the scope of this research in yellow)

1.5 Terminology

"Cartographic visualisation", "geovisualisation", "visualisation of geo-spatial data" and "map-based visualisation" are amongst the terms used to describe information visualisation involving maps. The terms geovisualisation and cartographic visualisation are generally favoured in cartographic research (MacEachren and Kraak 2001; Reichenbacher 2004). The focus of this research is on systems which make use of maps to provide useful visualisations of information. As a result, the more generic term, "map-based visualisation", is used in this dissertation. Terms such as geovisualisation are still used when referring to the names of algorithms, models or frameworks.

The term "mobile" in mobile map-based visualisation is used to refer to all hand-held mobile devices, including smart phones, personal digital assistants (PDAs) and mobile navigation devices. The problems of mobile visualisation and MMV are universal among these devices. This research will largely focus on mobile phones and especially those devices that do not provide for touch-screen input, as the problems of MMV are most severe on such devices. The terms "mobile device" and "mobile phone" are used where appropriate in this dissertation.

An "adaptive MMV system" is an MMV system that improves interaction with the user by constructing a user model based on observed user behaviour (Langley 1999). Additional models may also inform the adaptation process, but the user model remains of primary importance.

1.6 Research Questions

This research will aim to answer several research questions in order to address the situation of concern identified in Section 1.2. These research questions are reflected in the structure of this dissertation and mirror the objectives outlined in Section 1.3. Specifically, these research questions are the following:

1. What are the problems and shortcomings of existing MMV systems and techniques? (Chapter 2)
2. How can AUIs address the problems associated with MMV? (Chapter 3)
3. How can a model for adaptive MMV systems be designed? (Chapter 4)
4. How can a prototype based on the proposed model be implemented? (Chapter 5)
5. What benefits does this prototype provide? (Chapter 6)

The appropriate method(s) to address the above research questions are discussed in the following section.

1.7 Methodology

Several methods were used in order to answer the above research questions. These are summarised below.

1. What are the problems and shortcomings of existing MMV systems and techniques?

In order to identify the problems and shortcomings associated with MMV, a literature study of mobile visualisation and MMV was conducted. The problems associated with mobile visualisation and MMV are well-documented. Existing literature was analysed in order to identify these problems. A wide variety of techniques which attempt to address the problems and shortcomings of mobile visualisation and MMV have previously been proposed. These techniques were categorised and analysed in order to determine whether they adequately address the problems and shortcomings of MMV.

2. *How can AUIs address the problems associated with MMV?*

A literature study was conducted to identify how AUIs can benefit MMV systems and the potential they provide for addressing the problems and shortcomings identified in the literature study of MMV. The theory of AUIs has been developed over the last two decades and was examined to define AUIs in the context of this research and to identify possible benefits which AUIs can provide. Several well-known problems and shortcomings inherent to AUIs were investigated. The literature study also allowed the major components of AUIs to be identified, as well as the roles these components play in the adaptation process. Existing research into the use of AUIs in MMV systems was reviewed, to identify how AUIs have already been used to benefit MMV systems. The major input variables and the types of adaptation that have been performed were identified.

3. *How can a model for adaptive MMV be designed?*

Several relevant models, frameworks and architectures were identified and reviewed. An analysis of these models was conducted in order to determine which of the major input variables and adaptation effects were supported. This analysis revealed several shortcomings, motivating the need for a new model to be developed to allow the design of MMV systems which are able to achieve the full benefits of AUIs.

The model proposed in this dissertation was designed to address the shortcomings of existing models. The design of the model was based on research examined during the literature study of AUIs and supports the input variables and adaptation effects identified in existing research. The proposed model builds on these existing models, incorporating additional components (where necessary) to address the shortcomings identified.

4. *How can a prototype based on the proposed model be implemented?*

A proof-of-concept prototype was implemented based on the proposed model. This was done in order to determine whether the design of adaptive MMV systems based on the model is feasible. Olivier (2004) identifies four reasons for implementing prototypes. Two of these are particularly relevant for this research. Firstly, the prototype was implemented in order to demonstrate the feasibility of the model. Secondly, the prototype provided a useful tool for experimentation and was useful in answering the last research question.

5. What benefits does this prototype provide?

The proposed model was designed to facilitate the development of MMV systems which make use of AUIs to address the problems of MMV. In order to evaluate whether the model achieved this objective, usability evaluations of the prototype were conducted. Both a preliminary usability evaluation and a longitudinal field study were conducted. The motivation for the use of these techniques and the experimental design of these evaluations is discussed in more detail in Chapter 6. The evaluations were undertaken in order to determine whether the adaptations implemented in the prototype provide significant benefits for an MMV system.

1.8 Significance

AUIs provide many potential benefits for MMV systems. Adaptation of the information, visualisations and user interface of MMV systems provides the potential to alleviate many of the problems associated with MMV. By developing and evaluating an adaptive MMV system, this research will allow us to determine whether adaptation does indeed help to address the problems associated with MMV.

The model proposed in this dissertation will address the shortcomings of existing research into the use of AUIs in MMV systems. Despite the existence of several models, frameworks and architectures for the design of adaptive MMV systems, no model exists which incorporates all of the input variables and classes of adaptation effects to be found in literature. As a result, many adaptive MMV systems are created which implement adaptation in an ad-hoc manner, making the benefits achieved difficult to replicate. The proposed model will allow the benefits of model-based design to be realised in adaptive MMV systems, by serving as a template for the design of such systems in the future.

1.9 Dissertation Outline

The literature study of MMV is presented in Chapter 2. The relationship between MMV and information visualisation, map-based visualisation and mobile visualisation is examined. The benefits of mobile visualisation and the tasks supported by MMV systems are identified. Problems and shortcomings associated with mobile visualisation and MMV are discussed and analysed. Existing techniques which have been used to address the problems of MMV are discussed and classified according to the different approaches used. The shortcomings of existing approaches are highlighted, motivating the need for the use of AUIs.

AUIs are discussed in Chapter 3. The relationship between intelligent user interfaces and AUIs is described and AUIs are defined. The benefits provided by AUIs are identified, followed by a discussion of the problems and shortcomings associated with AUIs. Typical AUI components are identified and the major classes of components are described in more detail, with a particular focus on AUI techniques that have been applied in mobile and MMV systems. Existing research into the use of AUIs in MMV systems is examined and different classes of input variables and adaptation effects identified. Finally, existing models, frameworks and architectures relevant to this research are discussed and compared in terms of the input variables and adaptation effects supported. The analysis of the shortcomings of these existing models, frameworks and architectures is used to motivate the need for the development of a new model for adaptive MMV systems.

A new model is proposed in Chapter 4 to address the shortcomings of existing models. This model builds upon the existing models, frameworks and architectures discussed in Chapter 3. Modifications are made to address the shortcomings of existing models. The proposed model describes the design of adaptive MMV systems from several perspectives. The model architecture is presented, followed by a more detailed discussion of the individual components. The interfaces between the components, data design and algorithm design are also described.

A proof-of-concept prototype was developed based on the proposed model. Chapter 5 describes the design and implementation of this prototype. The implementation of the architecture, individual components, data design, interface design and algorithm design is described. The different forms of adaptation facilitated by the prototype are also described.

The evaluation of the prototype is discussed in Chapter 6. The choice of usability evaluation techniques is motivated. Two usability evaluations are then described, firstly a preliminary evaluation to assess the usability of the prototype and secondly a longitudinal international field study. The experimental design of each evaluation is discussed, followed by the presentation and analysis of the results, categorised into performance and satisfaction results.

Conclusions that can be drawn from this research are discussed in Chapter 7. This chapter examines whether the objectives of this research were achieved and highlights the theoretical and practical contributions made. Recommendations for theory, practice and future research stemming from this research are also made.

Chapter 2: Mobile Map-based Visualisation

2.1 Introduction

This chapter is the first of two literature study chapters used to inform the design of the proposed model for adaptive MMV systems. The primary purpose of this chapter is to identify the problems and shortcomings of existing MMV systems and techniques. This chapter begins with a discussion of the basic theory behind information visualisation (IV) in general, specialising into map visualisation, mobile visualisation and mobile map-based visualisation (MMV). Benefits and problems of mobile visualisation in general and MMV in particular are identified. Existing techniques for MMV are then categorised into the different approaches that currently exist and analysed to determine how well they address the challenges of MMV. Existing MMV systems are used for illustrative purposes where appropriate.

2.2 Information Visualisation

IV has been defined as:

> "*The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.*" (Card *et al.* 1999)

Visualisation pre-dates computers. For centuries, mankind has devised means of visualising data in order to make it easier to gain insight into the phenomena it describes. It is only in the last fifteen to twenty years that the rapid growth in the data generation and storage capabilities of computers has fuelled the need for improved means of exploring this data and extracting useful information (Spence 2007). The growth in memory capacity has been accompanied by advances in graphical output technology and processing power, empowering new and improved techniques for visualisation. Today, a wide range of visualisation techniques exist for the visualisation of a wide variety of data sets, ranging from simple textual data to complex multidimensional data sets. As the technology continues to develop, new challenges and opportunities are emerging in the field of IV.

IV tools allow users to explore and interact with data sets. Shneiderman (1996) identifies seven high-level tasks involved in IV, as follows:

- *Overview:* View the entire data collection.

- *Zoom:* Zoom in to items or areas of interest, or zoom out to get a higher-level view.
- *Filter:* Remove uninteresting items by filtering according to certain criteria.
- *Details on demand:* Select an item to view details for that particular item.
- *Relate:* View relationships between items.
- *History:* Store user actions to allow undo/redo of actions.
- *Extract:* Allow views and associated query parameters to be saved.

Shneiderman maintains that overview, zoom, filter and details on demand are the core tasks involved in information exploration and that these tasks should be supported by any IV tool.

Several models and frameworks exist which describe the IV process, including the Visualisation Reference Model (VRM) (Figure 2.1), one of the simplest and best-known examples (Card *et al.* 1999). The Information Visualization Data State Reference Model is a very similar model (Chi and Riedl 1998). The VRM splits the visualisation process into four distinct phases, transforming raw data into the visual form that is output to the user. These four phases are:

- *Raw data:* Data in unprocessed form, as recorded or read from some input source.
- *Data table:* Data in relational form, including metadata. Data tables are more structured than the raw data, allowing the data to be more easily represented in visual form. Examples of metadata include row and column names for the data table in question. Data tables consist of sets of tuples, with each tuple containing values for different attributes.
- *Visual Structures:* Data tables are mapped onto visual structures which are appropriate for displaying the data. For example, two dimensional data might be mapped onto a scatter plot or map. Selecting the appropriate visual structure is important as it can affect how easily the visualisation can be understood. Selecting the incorrect visual structure can create incorrect perceptions regarding the underlying data and lead to misinterpretation of the data.
- *Views:* Users are normally able to manipulate the underlying visual structure in order to visualise the data as they wish. This could involve selecting an item to view details

on demand, manipulating the viewpoint through zooming and panning, or applying some sort of distortion technique (e.g. fisheye lens) to manipulate the view.

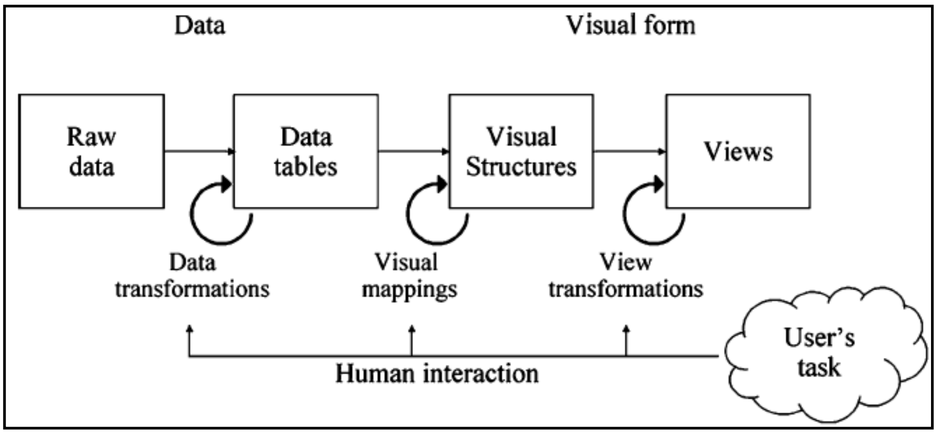

Figure 2.1. Visualisation Reference Model (Card *et al.* 1999)

The VRM also identifies three areas for human interaction with the visualisation:

- *Raw Data to Data Table (Data Transformations):* This could be as simple as selecting a subset of the raw data to visualise (using a filter control) or organising the underlying data into groups.
- *Data Table to Visual Structure (Visual Mappings):* The user can often manipulate the visual mapping between the data table and the visual structure. For example, if the visual structure in use was a bar chart, the user could change the variable used as the x-axis label.
- *Visual Structure to View (View Transformations):* The view can often be manipulated by zooming and panning or selecting a display element to view details on demand.

A wide variety of data types have been visualised and map-based data is amongst the most common.

2.3 Map-based Visualisation

IV taxonomies typically categorise data according to its dimensionality. Shneiderman (1996) identifies seven major data types that are typically found in IV systems, namely 1D, 2D, 3D, multi-dimensional, temporal, network and tree data. In this taxonomy, map-based visualisation falls into the 2D category, with latitude and longitude being the two dimensions. Spence (2007) classifies "geovisualisation" separately from IV, classifying visualisation into IV, scientific visualisation and geovisualisation. He distinguishes geovisualisation from IV because geographic or map-based data is often not abstract as the definition in Section 2.2 states, but is usually linked to a specific physical location. In this sense, geovisualisation is a hybrid of IV (which models abstract data) and scientific visualisation (which models physical or observable phenomena). Despite this distinction, many of the techniques developed for IV in general have been applied to map-based visualisation and all seven of Shneiderman's IV tasks are supported by map-based visualisation systems.

Map-based visualisation is not limited to maps in the traditional sense of the word. Several other data types have been visualised using map metaphors, as map-like interfaces often provide a useful visual structure for visualising other data types. Non-map data has been mapped onto 2D or 3D maps to allow users to gain insight into underlying patterns in the data (Skupin 2000). Multi-dimensional data has also been mapped onto lower dimensional Self-Organising Maps (SOM) (using machine learning techniques such as neural networks), which are easier to visualise (Engelbrecht 2002).

Map-based visualisation using computers was originally largely restricted to Geographic Information Systems (GIS). A GIS consists of a database management system, a set of operations for exploring data and a graphic display system used to analyse geo-spatial data (Rhyne and MacEachren 2004). It is in this graphic display system that map-based visualisation plays an important role. GIS systems were originally the domain of expert users, running on expensive hardware (Reichenbacher 2004). Recently, the Internet has stimulated the development of a number of successful map-based systems for the mass market, with Google Maps[4] and Google Earth[5] being among the most successful and well-known (Kraak 2006). Similar systems have also emerged for mobile devices (both phones and specialised GPS navigation devices). Map-based visualisation systems are now available to the mass market, increasing the need for systems to be developed which are easy to use.

MMV is examined in more detail in Section 2.5. Mobile visualisation in general is discussed in the following section.

[4] http://maps.google.com

[5] http://earth.google.com

2.4 Mobile Visualisation

The first commercial mobile phone network based on cellular network technology was launched in Japan in 1979 (King and West 2002). Since then, mobile phones have changed dramatically. Thirty years ago, phones were bulky and had no graphical displays. In the early to mid 1990s, this started to change, with phones becoming smaller and small displays of low quality becoming commonplace on second generation, or 2G phones. In the last decade mobile phones have developed into viable visualisation platforms. Today, mobile phones are not just used for communication but have evolved into platforms for the delivery of information and services. Other mobile devices, such as mobile GPS navigation devices, are now also commonplace. High-end mobile phones today have secondary storage capacities of many gigabytes, processors with clock speeds of hundreds of megahertz and up to hundreds of megabytes of RAM. This is roughly equivalent to the capabilities of desktop computers less than ten years ago. Mobile phones are now also equipped with larger and better quality displays than in previous years. Mobile phones today thus have the data storage, processing and graphical output capabilities to be used as visualisation platforms.

2.4.1 Benefits

Mobile devices provide several advantages over desktop systems as visualisation platforms. Firstly, mobile devices are typically carried everywhere by users. As a result, mobile visualisation systems are useful for providing real-time visualisations to users on the move. Secondly, mobile devices provide a number of sensors that can be used to determine the user's context. Context aware computing is a major research area on its own which provides important potential benefits for mobile visualisation.

Context information can be defined as follows:

> "*…any information which characterizes the situation of an entity that is relevant to the usage situation*" (Dey *et al.* 2001).

Several sensors are available on many mobile devices to determine the user's context. Location sensors are among the most widely used, but other sensors are also increasingly being used, including accelerometers, proximity sensors and ambient light sensors (Schmidt *et al.* 1999). The mobile device's location can be determined in different ways, with differing levels of accuracy. Cell ID location determines the location of a mobile phone from the position of the base station with which the phone is communicating (Openwave 2002). This is usually accurate to within three kilometres (depending on the cell size), as mobile phones normally measure the signal strength from the six nearest base stations and communicate with the station with the strongest signal. This results in each base station dominating a roughly circular region which can be used to approximate the user's position.

A variety of techniques can be used to improve on basic Cell ID positioning and improve its accuracy. Figure 2.2 shows a screenshot of Google Maps for Mobile[6], showing the use of Cell ID positioning to approximately determine the user's position. Built-in GPS receivers are also becoming more common in mobile devices. These GPS receivers are usually very weak and often rely on Assisted GPS (A-GPS) to improve accuracy. A-GPS makes use of assistance servers, with which the mobile device communicates to help determine its position when it is not able to do so autonomously (Rao and Minakakis 2003).

Figure 2.2. Google Maps for Mobile showing user location determined using Cell ID

The ability to sense the user's context offers a wide range of potential benefits for MMV systems. Location-based services, which provide information and services relevant to the user's location, can be provided (Zipf 2002). Systems can also adapt in a variety of ways to take advantage of knowledge regarding the user's context. The use of context to adapt MMV systems will be discussed in Chapter 3.

---

[6] http://www.google.com/gmm

2.4.2 Problems and Shortcomings

Mobile devices provide unique opportunities to deliver visualisations to users on the move. These advantages, however, come at a price, as some sacrifices have to be made in comparison to desktop visualisation. Visualisation techniques have to be adapted in order to accommodate device limitations, including limited screen size (Karstens *et al.* 2003).

Several factors have been identified as shortcomings to be considered when developing visualisation systems for mobile devices as opposed to desktop computers. These can be classified into three categories, namely resource-related problems, device-related problems and problems related to the context of use of mobile devices (environmental problems). These shortcomings are summarised in Table 2.1 (Björk *et al.* 2000; Chittaro 2006; Pattath *et al.* 2006).

Category		Problem/Shortcoming
Category	**Resource**	• CPU, memory, buses and graphics hardware are less powerful; • Limited power supply and battery life; and • Low network bandwidth and unreliable connectivity.
	Device	• Smaller displays, lower resolutions, fewer colours; • Input peripherals make performing complex tasks difficult because of limited size and unwieldy nature; and • Capabilities and input peripherals differ greatly between different devices.
	Environmental	• Variability in the visual environment including differences in lighting conditions which may affect the perception of colour; • The auditory environment may hamper the use of sound as output; • The mobile visualisation system may not deliver the performance required in a particular mobile context; and • Use of a mobile device is often a secondary rather than a primary task as the user may be busy doing something else.

Table 2.1: Problems and shortcomings of mobile visualisation

Some of these problems will gradually be solved as the hardware capabilities of mobile devices improve. Other problems, such as limited screen size will remain, simply because a small screen is necessary for a mobile device to be practical.

2.5 Mobile Map-based Visualisation

It has been suggested that 85% of all data has a spatial component (MacEachren and Kraak 2001). It is therefore not surprising that many of the most popular mobile visualisation applications incorporate map-based visualisations. MMV shares the benefits and shortcomings of mobile visualisation in general. This class of system does, however, introduce its own unique problems and shortcomings that need to be addressed. This section begins by examining the different applications of MMV that currently exist. The different high-level tasks supported by MMV systems are also identified. Finally, problems and shortcomings that are relevant to MMV are discussed.

2.5.1 Existing Applications

A large proportion of existing MMV systems are map-based mobile guides (Baus *et al.* 2005). Many of these are targeted at tourists, providing support for tasks such as navigation and locating tourist attractions (O'Grady and O'Hare 2004). Other systems target a wider range of users, supporting users in route planning, providing real-time traffic updates and identifying points of interest such as restaurants and shops (Bornträger *et al.* 2003; Alimohideen *et al.* 2006; Andriakopoulou *et al.* 2007; Nokia 2008b). Another important application of MMV systems is for use by emergency services (Betts *et al.* 2005). This application was further stimulated by legislation in the US requiring that the emergency services be able to identify the location of any mobile device. The Federal Communications Commission (FCC) issued a mandate called the Enhanced 911 (E911) mandate requiring wireless operators to be able to provide the location of a device automatically to the police or other emergency rescue services (FCC 2003). A wide variety of other applications have been suggested, including supporting field engineers in addressing problems in a Wide Area Network (WAN) (Mitchell *et al.* 2005), management of natural resources and business applications (Burigat and Chittaro 2008).

2.5.2 Tasks

Existing MMV systems support a wide range of common user tasks. Four high-level tasks associated with the visualisation of geographical data have been distinguished, namely identify, locate, compare and associate (Knapp 1995). Similarly, a set of four tasks commonly supported by mobile tourist guides has also been identified (Von Hunolstein and Zipf 2003). These four tasks are:

- *Locator:* Identify the position of something (e.g. where am I?)

- *Proximity Tasks:* Identify nearby facilities (e.g. where is the nearest petrol station?)
- *Navigation Tasks:* Find the route to a particular location.
- *Event Tasks:* Determine the condition of a particular location (e.g. operating hours of a business).

Reichenbacher (2004) provides a taxonomy of tasks supported by MMV systems that has a lot in common with the above set of tasks. The *Locator* and *Proximity* tasks are combined into a single category, called *Orientation and Localisation.* The *Navigation* and *Event* tasks are also included. Two additional tasks are also identified, namely:

- *Search:* Search for a person or object that matches certain search criteria (e.g. list petrol stations within 5km of my current position).
- *Identification:* Identify or recognise people or objects. This usually involves providing semantic information (e.g. the name of a location on a map).

These actions can be combined into a particular activity, e.g. going to a movie can combine locating, navigating, searching and event checking.

2.5.3 Problems and Shortcomings

In Section 2.4.2 a number of problems and shortcomings of mobile visualisation in general were identified. All of these problems are relevant to MMV systems, with some being further aggravated by the type of data visualised in MMV systems. The following problems and shortcomings of MMV systems have been identified (Reichenbacher 2004; Burigat and Chittaro 2005):

- Maps are usually far larger than the 240 x 320 pixel displays typically provided by most advanced mobile phones. This often results in the user having to scroll long distances, either indirectly manipulating the map through a key-pad, or requiring both hands, as is the case with touch-screen devices. Zooming and scrolling can become tedious if considerable interaction is required to find information of interest. Users may also experience problems keeping track of how the current view fits into the overall context, especially at high zoom levels. When zoomed out, details are often too small to distinguish.
- Many of the typical tasks users perform with MMV systems are interaction intensive (Section 2.5.2). For example, in performing a navigation task, users may be required

to first enter various search criteria to find their destination. Users may then have to set preferences for the route to be planned, as well as the starting point and destination. Finally, zooming and panning operations may be required for users to get their bearings while navigating to their destination. Such interaction intensive tasks further aggravate the problems associated with the input peripherals available on mobile devices.

- MMV systems either render maps in real-time (in which case the limited processing power and memory capacity are problematic) or download maps from the Internet (in which case limited network bandwidth and unreliable connectivity are of concern).
- Maps used by MMV systems are usually borrowed from desktop systems, or are scanned versions of paper maps. As a result, these maps are often too detailed and unsuitable for use in MMV systems.
- The small screen of mobile devices means that there is often not enough space for auxiliary elements such as a map legend, resulting in the visualisations being difficult to interpret. There is also usually limited or no support for an overview function, making it difficult for the user to see how the current view fits into the overall context.
- The fact that users are mobile means that they could potentially be dividing their attention between the system and another activity. MMV systems are often used as aids to complete an activity (such as navigating to a destination).
- Clutter can be a significant problem in MMV systems where a large amount of information must be visualised in a very small display area. Clutter can be defined as *"the state in which excess items, or their representation or organization, lead to a degradation of performance at some task"* (Baudisch and Rosenholtz 2003).

In the preceding sections a number of problems and shortcomings of mobile visualisation and MMV were identified. In the following section, existing mobile visualisation and MMV techniques will be discussed and analysed to determine to what degree these techniques address the problems and shortcomings identified.

2.6 Existing Techniques

The discussion of existing techniques will focus on MMV techniques, as these are most relevant to the scope of this research. Where appropriate, techniques used in map-based visualisation or mobile visualisation in general will also be discussed.

Desktop visualisations cannot merely be scaled down for mobile devices. Special compensation is required to address the limitations of these devices and ensure that mobile visualisation applications are able to run efficiently and without serious usability problems (Chittaro 2006).

Three aspects which should be considered when designing mobile visualisation systems have been identified (Karstens *et al.* 2003):

- Reducing the computational complexity of visualisation techniques;
- Addressing the limited screen size; and
- Adapting the user interface.

Techniques for addressing limited screen size will receive the most attention in this chapter. Reducing computation complexity can be fairly easily achieved by compromising on the quality of the visualisations rendered. Using simple graphics primitives, using 2D effects rather than 3D where possible, reducing use of processing-intensive graphical effects (such as transparency and gradient paints) and using fewer colours can all help to reduce the computational load of such systems. Adapting the user interface is a more complex topic and will be addressed in Chapter 3, as an understanding of the theory of AUIs is required before the use of such interfaces in mobile visualisation and MMV can be discussed.

Addressing the limited screen size is a very serious problem in mobile visualisation and is referred to as the presentation problem (Chittaro 2006). MMV systems typically visualise large information spaces on a small screen and have to find ways of dealing with the presentation problem. Several techniques have been used to attempt to overcome this problem, with varying levels of success. The problem of displaying data which does not fit on a single screen is not new to Human-Computer Interaction (HCI) research and many of the techniques employed in mobile visualisation were originally developed for desktop systems (Gutwin and Fedak 2004). Existing techniques can be categorised into the following five categories, namely hierarchical organisation, zooming and panning, overview + detail, focus

+ context and reference to off-screen objects. These techniques are discussed in more detail in the following sections.

2.6.1 Hierarchical Organisation of Information

One of the simplest techniques for displaying information that is too large for the small screen is to arrange it into an information hierarchy (Sanna and Fornaro 2003). In this way, the information can be split into portions of manageable size, each of which is small enough to fit onto the display. The different portions are organised into a hierarchy which enables the user to find information of interest. This helps to minimise or even eliminate the need for users to scroll the display to locate information. This technique does, however, have two significant disadvantages. Firstly, a lot of interaction may be required in order to find information of interest. Given the limited interaction techniques provided by mobile devices and the potential to select the wrong split in the hierarchy at any point, this could lead to a frustrating user experience, with the user having to move up and down the hierarchy. Secondly, users may lose sense of their current position in the hierarchy. For example, consider Figure 2.3, which shows the P.I.V.I intrusion detection visualisation system (Sanna and Fornaro 2003). No indication is given as to where the current visualisation falls in the hierarchy. Only two buttons, “Level Up” and “Select”, are provided to navigate the hierarchy.

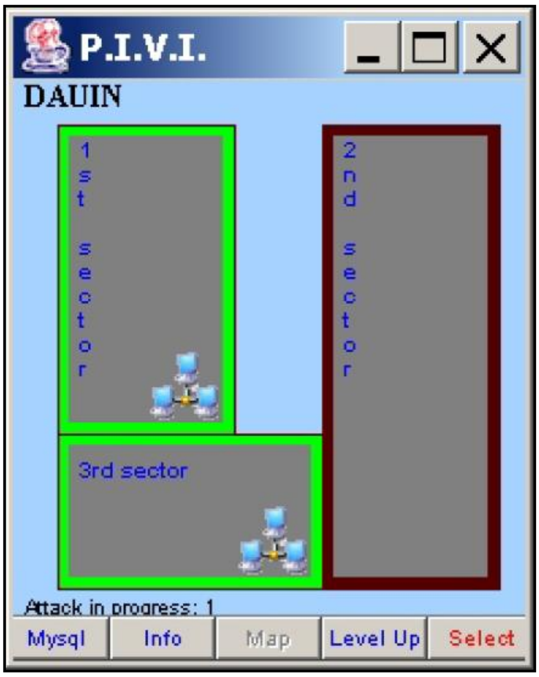

Figure 2.3: Hierarchical organisation in an intrusion detection system (Sanna and Fornaro 2003)

Not all information can be organised into a hierarchy. An alternative approach has been proposed, using Radial Edgeless Trees (RELTs) to maximise use of the mobile device display area by recursively splitting the available screen space in order to display hierarchical information (Hao and Zhang 2007). Hierarchical organisation is rarely used in MMV systems, as spatial data is not generally suited to hierarchical organisation.

2.6.2 Zooming and Panning

Another simple technique used for navigating large visualisations on small displays is zooming and panning. In this technique, only a portion of the visualisation is visible at a time. Users are able to explore the visualisation by shifting the viewing frame to the desired location (panning) and zooming in or out, using either fixed or user-controlled levels of magnification (Gutwin and Fedak 2004). One problem with this approach is that users can lose track of the global context when zoomed in on a detailed view, while details can be difficult to distinguish when zoomed out (Chittaro 2006). The two visualisation techniques which follow (overview + detail and focus + context) attempt to overcome this problem by making the user aware of how the detailed view fits into the global context. Zooming and panning has proved popular with MMV systems, as it is easy to implement and represents a natural way of interacting with the large 2D information spaces typical of such systems.

Several variations on simple zooming and panning have been developed. Zoomable User Interfaces (ZUIs), using toolkits such as PAD++ (Bederson *et al.* 1996) (later updated to create the Piccolo toolkit), allow large information spaces to be displayed on mobile devices by allowing the user to zoom in on a portion of the display. The difference between ZUIs and traditional zooming and panning interfaces is that transitions between the different zoom levels are animated, helping the user to maintain a mental picture of where the current view fits into the global context (Hakala *et al.* 2005).

Zooming and panning techniques can be frustrating to use on mobile devices, where the user may have to perform a significant amount of panning in order to navigate large information spaces (Kaptelinin 1995). Consider the case of a mobile navigation system, where the user has to plan a route from one town to another. If no shortcuts are available to skip to the desired destination, a significant amount of interaction could be required to successfully complete this task. Speed-dependent Automatic Zooming (SDAZ) attempts to address this problem. SDAZ links the zoom-level to the speed at which the user is scrolling across the display. For example, if the user wants to rapidly scroll across a large area, the zoom level is automatically decreased, allowing the distance to be traversed more quickly. This technique has been shown to help users complete scrolling tasks quicker than traditional zooming and panning approaches (Cockburn and Savage 2003). Favourable results have also been achieved using SDAZ in an MMV system (Jones *et al.* 2005).

Gutwin and Fedak (2004) introduced another variation on zooming and panning techniques, through the use of a two-level zoom approach. Users can switch between an overview and a fully zoomed-in view, which can then be navigated by panning. This is intended to reduce unnecessary panning on the part of the user. The potential shortfall of this technique is that it depends to a large extent on whether the user can distinguish sufficient detail in the zoomed-out overview for it to be of any use.

Interacting with visualisations of the zooming and panning variety can be problematic on mobile devices which do not provide for touch-screen interaction. The ZoneZoom system (Robbins *et al.* 2004) provides a means of addressing this problem. ZoneZoom divides the display into a grid and maps the different cells of the grid onto keys on the keypad of a mobile phone (Figure 2.4). Rosenbaum and Schumann (2005) expand on this to allow zooming and panning on the currently selected image portion.

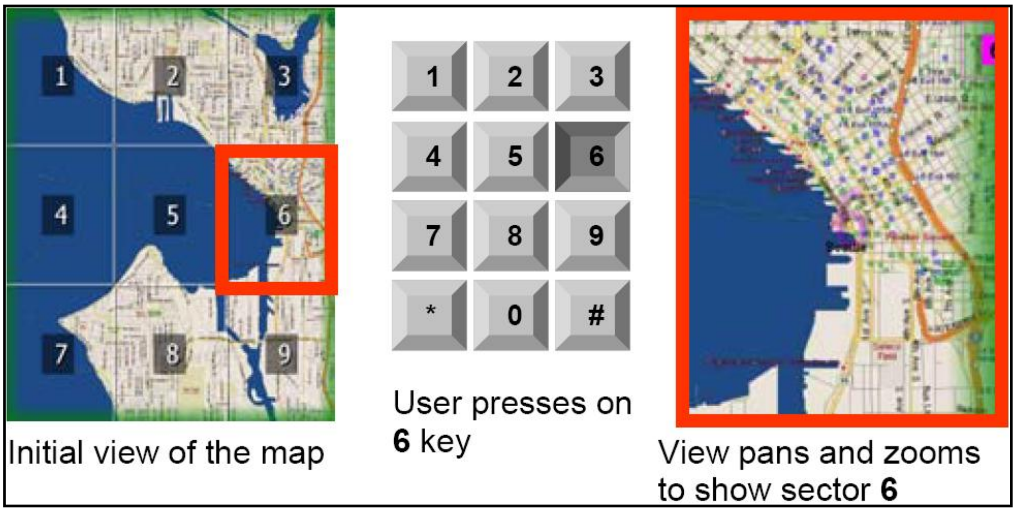

Figure 2.4: Using a mobile phone keypad to zoom into a portion of the current view (Robbins *et al.* 2004)

Zooming and panning interaction techniques vary, depending on the type of input peripherals available. On mobile devices without touch screens, panning is usually performed using the cursor keys and zooming through the use of dedicated zoom-in and zoom-out keys. On touch screen devices which rely on a stylus for input, interaction techniques are more varied. Panning can be performed by drawing the stylus across the screen to “drag” the view, tapping on the edge of the screen to move in that direction or tapping the display to centralise the view on that point (Looije *et al.* 2007). Zooming techniques include the use of on-screen controls, dragging upwards or downwards, or selecting a rectangular region to be magnified. Interaction through moving the device itself (and detecting motion using the phone's camera), tilting of the mobile device and the use of sound have also been proposed as techniques for interacting with MMV systems (Gutwin and Fedak 2004). Projectors built into mobile devices have also been proposed to enlarge maps to make interaction easier (Hang *et al.* 2008).

2.6.3 Overview + Detail

Overview + detail represents a well-established and proven technique in desktop visualisation systems (Plaisant *et al.* 1996; Kumar *et al.* 1997; Card *et al.* 1999). This technique attempts to overcome the problems associated with zooming and panning by dedicating a portion of the display to an overview which shows how the current detailed view fits into the overall picture. This allows the user to examine a detailed view, while still being aware of how that detailed view fits into the global context. Burigat *et al.* (2008) extended the basic ZUI approach, to include an overview in the bottom right corner of the display to give the user an idea of how the current view fits into the broader context. A screenshot of their "Classic ZUIO" approach is provided in Figure 2.5, showing the use of semantic information in the overview (the location of points of interest is denoted by the use of green and purple highlights) in order to aid users in navigating to these points on the map. Their evaluation of this technique showed it to be beneficial to the user when searching for items in a map-view, with the amount of zooming and panning required being significantly reduced.

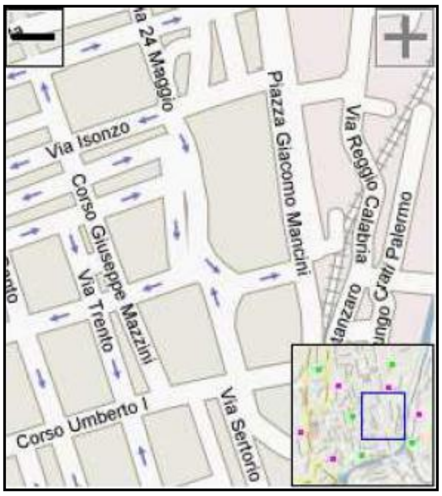

Figure 2.5. Classic Zoomable User Interface with Overview (ZUIO) (Burigat *et al.* 2008)

Figure 2.6 shows another example of the use of overview + detail in an MMV system for visualising points of interest (POIs). Crosshairs on the overview in the lower right corner of the display indicate the location of the currently displayed detailed view within the broader context. This example clearly shows the problems with this technique when applied to mobile devices. While the overview assists the user by showing where the current visualisation fits into the global context, it is too small to distinguish meaningful information, which may detract from its usefulness. It also obscures part of the detailed view. As a result, this technique does not scale well to mobile devices, making it unsuitable for MMV systems.

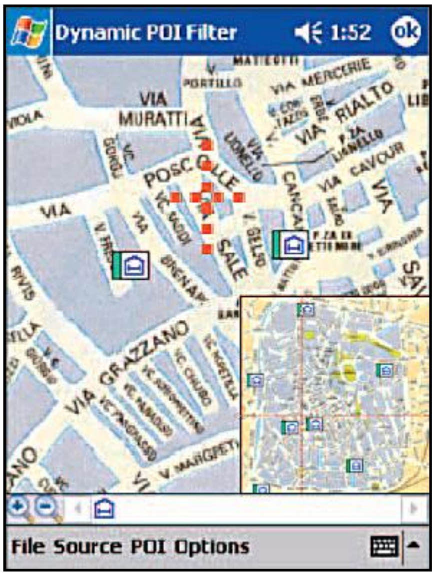

Figure 2.6: Overview (bottom right hand corner) + Detail Visualisation (Chittaro 2006)

2.6.4 Focus + Context

Focus + context follows a similar approach to the overview + detail technique in that it recognises that users need access to a detailed view, while retaining some idea of how that detailed view fits into the global context. Focus + context differs from overview + detail in that it does not separate the overview and detailed views. Instead, both are integrated into a single view, with the detailed information dominating the display and related information on the periphery in less detail (Card *et al.* 1999). Techniques which aim to provide focus and context information in a single view usually make use of some form of distortion, with a portion of the screen magnified and information on the periphery compressed. Previous focus + context visualisations have used a variety of techniques including fisheye views (Furnas 1986), Flip Zooming (Björk *et al.* 2000) and the TableLens (Rao and Card 1994). Other techniques, such as Focus Maps (Zipf and Richter 2002) have been developed specifically for

MMV systems. Fisheye views and Focus Maps relate directly to MMV and are discussed in the following sections.

2.6.4.1 Fisheye Views

Fisheye views were first proposed by Furnas (1986) as a means of addressing the simultaneous need for local detail and global context when dealing with a large information space. A fisheye lens is used to display a portion of the visualisation (often the centre) in focus, while compressing surrounding areas. The compression of the periphery saves on space and allows overview and detail to be integrated into a single display.

Figure 2.7. Use of the fisheye technique in an MMV system (Spence 2007)

Fisheye views have rarely been used for mobile devices (Bederson *et al.* 2004). It has been suggested that this is because it would be difficult and inefficient to implement such a mechanism on a mobile device (Chittaro 2006). Fisheye views have, however, been applied in a calendar application for PDAs called DateLens (Bederson *et al.* 2004). DateLens allows a portion of the calendar (such as the current day) to be magnified, with surrounding days compressed on the periphery. Evaluation results showed that the fisheye calendar produced better performance for more complex tasks. Fuchs *et al.* (2006) demonstrated the use of a similar simplified fisheye distortion in a mobile visualisation system to support air conditioning unit maintenance workers. They used a rectangular focus area of constant zoom, rather than the classic continuous distortion technique, in order to improve performance on mobile devices.

Fisheye views have also been used in MMV systems. Figure 2.7 shows the use of fisheyes in an MMV system visualising transportation links. Despite the distortion that occurs as a result of the differing scales in use inside and outside the fisheye, continuity is maintained so that no breaks in the transportation links occur. Figure 2.8 shows a similar approach, where different scales are used, depending on the proximity of the area of the map to the user's current location (Harrie *et al.* 2002).

Figure 2.8. Map using different scales for different regions of the map (Harrie *et al.* 2002)

2.6.4.2 Focus Maps

Focus Maps is a technique that was developed to ease the screen clutter problem often associated with MMV (Section 2.5.3) and to make map reading easier. This technique divides a map into different "zones", based on what the system regards as regions of interest to the user. Such regions of interest usually consist of the user's current location and destination (if a navigation task is involved). Two techniques are then used to draw the user's attention to the regions of interest, namely generalisation and use of colour. Zones that are considered to be of significance to the user are rendered in detail, with little generalisation. Less interesting zones are rendered with increasing generalisation, inversely proportional to the level of interest. High interest zones are rendered with bright colours, while dull colours are used for low interest regions. In this way, users' attention is drawn to the salient areas and information on the map, given their current task (Zipf and Richter 2002). An example of the Focus Maps technique is shown in Figure 2.9.

Figure 2.9. Focus Maps example showing the same map with (right) and without (left) Focus Maps (Zipf and Neis 2007)

2.6.5 Visualising Off-Screen Objects

Section 2.6.2 discussed zooming and panning techniques and the associated problem that no indication is provided to the user of how the current view fits into the broader context. Focus + context techniques overcome this to some degree, but can be undesirable in situations where the user needs to have a precise mental model of where objects not in focus are located. The use of focus + context techniques in MMV systems can be problematic, as the compression and distortion that takes place makes it impossible to determine the exact location of objects not in focus (Baudisch *et al.* 2002). As a result, standard zooming and panning techniques are generally preferred. Such techniques can be enhanced by providing indications to the location of off-screen objects. Three such techniques have been developed. Halo surrounds off-screen objects with circles just large enough to encroach onto the currently focused area, allowing the user to determine the distance and direction to off-screen objects (Baudisch 2004). CityLights uses points, lines or arcs on the border of the display to denote the direction and distance to off-screen objects (Zellweger *et al.* 2003), while arrows have been used by a variety of applications (including games) to achieve the same objective. Table 2.2 provides a summary of the methods used by the different techniques to visually encode information regarding off-screen objects. These techniques can be useful in helping to provide a sense of context to systems that rely on zooming and panning. They can, however, further clutter already crowded displays. Figure 2.10 shows examples of these three techniques in use.

		Encoding Method	
		Direction	**Distance**
Technique	**Halo**	Arc position	Arc curvature
	Arrows	Arrow orientation	Arrow length Arrow size Arrow colour Arrow shape Label
	CityLights	Line position	Line thickness Line colour Label

Table 2.2: Comparison of different techniques for visualising the location of off-screen objects (Burigat *et al.* 2006)

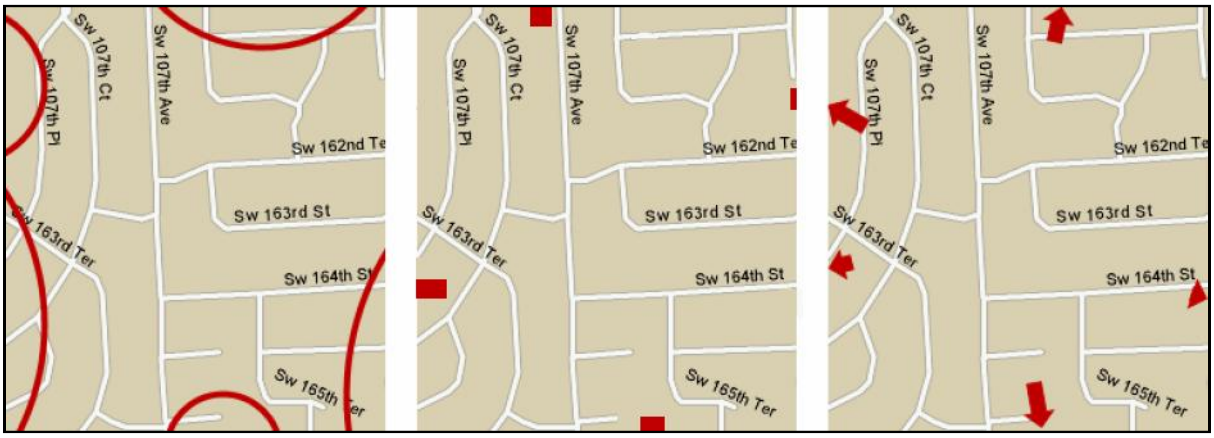

Figure 2.10. Three different techniques for visualising the location of off-screen objects. (From left to right) Halo, CityLights and Arrows (Burigat *et al.* 2006)

2.6.6 Comparison of Techniques

All of the above techniques come with associated advantages and disadvantages, as is illustrated in the summary provided in Table 2.3. While many of the techniques were developed specifically to solve problems associated with another technique, many of these techniques introduce new problems. Some studies have been done to compare the different techniques. Baudisch *et al.* (2002) compared focus + context, overview + detail and zooming and panning techniques, although in a desktop context. They found focus + context techniques to provide benefits in terms of efficiency and the ability of users to monitor and interact with the visualisation simultaneously. Gutwin and Fedak (2004) compared fisheye, two-level zoom (Section 2.6.2) and zooming and panning in a mobile context. Their findings stressed the importance of some sort of overview in improving efficiency and favoured the fisheye and two-level zoom techniques over zooming and panning.

Technique		Description	Advantages	Disadvantages
Technique	**Hierarchical Organisation**	Information organised into hierarchy. Split over several pages.	Little or no scrolling necessary.	May be difficult to find required information. Can lose track of global context.
	Zooming & Panning	Large information space navigated through zooming and panning.	Allows navigation of large information space without loss of detail.	Can lose track of global context.
	Overview + Detail	Display split into separate sections, showing detail and how it fits into global context .	Allows user to view detail and how it relates to the global context.	Overview too small to distinguish meaning. Obscures detail.
	Fisheye	Current view in focus, information on periphery compressed in proportion to distance from focus.	Detail and context combined into a single view.	Distance to peripheral objects is uncertain. Difficult to implement. Resource intensive.
	Focus Maps	Regions of interest to the user shown in greater detail and using brighter colours.	Addresses screen clutter problem by emphasising important information.	Complex and resource intensive.
	Visualising Off-Screen Objects	Distance and direction to objects not currently in view denoted by arcs, arrows or lines.	Allows user to keep track of off-screen information.	Can further clutter already cramped displays. Only partly addresses lack of overview.

Table 2.3: Comparison of mobile visualisation techniques

A variety of modifications have been made to standard desktop visualisation techniques in attempts to address the problems presented by mobile devices. The techniques discussed above, however, tend to focus on only one aspect of mobile visualisation and MMV. In most cases, the techniques attempt to address the problem of limited screen space and seek to provide the user with a sense of how the current view fits into the broader context. Some techniques, such as SDAZ and ZoneZoom (Section 2.6.2) seek to address the limitations of the interaction techniques available on mobile devices. The summary in Table 2.3 shows that existing techniques do not adequately address the problems of MMV. While many innovations have been introduced which provide advantages for MMV systems, these new techniques often introduce new problems. A more holistic approach is needed, which considers all aspects of MMV.

2.7 Conclusions

Information visualisation has proved useful for helping users understand and interact with a wide variety of data sources, transforming raw data into a visual form that users can explore. Map-based visualisation systems are among the most common and have evolved from specialised GIS to applications for the mass consumer market. Advancements in the capabilities of mobile devices mean that today such devices are able to serve as visualisation platforms with unique benefits in terms of mobility and context sensing capabilities.

Several MMV systems have been developed which provide a range of useful functions, from tourist guides to systems for assisting emergency service personnel. Despite these advantages, several problems face the designers and users of mobile visualisation systems. Limited screen space, processing power and memory capacity, as well as awkward interaction mechanisms are some of the problems that need to be overcome. Many of these problems are aggravated in MMV systems, which need to visualise a large information space which users can explore. A wide variety of techniques have been developed to address these problems, with the lack of screen space being a focus of much of the existing research. These techniques, however, tend to focus on only a single aspect of the problems of mobile visualisation and MMV. Furthermore, in addressing some of the existing problems, new problems are often introduced. A more holistic approach is needed to address these problems.

The use of adaptive user interfaces (AUIs) in MMV systems provides the potential to address many of the problems and shortcomings identified in this chapter. By adapting MMV systems to the preferences, context and tasks of the individual user, the possibility exists to create more comprehensive solutions to the problems identified.

A literature study of AUIs is presented in the following chapter, including a discussion of the theory behind AUIs and the application of AUIs to the problems of MMV. Existing models, frameworks and architectures for adaptive MMV systems are also discussed.

Chapter 3: Adaptive User Interfaces

3.1 Introduction

This is the second of two literature study chapters which will form the basis of the model for adaptive MMV systems to be presented in Chapter 4. The purpose of this chapter is to identify how AUIs can address the problems associated with MMV which were identified in the previous chapter. In order to do so, the basic theory of AUIs must first be discussed. In this chapter, AUIs are defined, common AUI components are identified and the functioning of an AUI is described. The use of AUIs in MMV systems is then discussed, in order to determine how AUIs have already been used in this domain. Finally, existing models and frameworks for adaptive MMV systems are discussed.

3.2 Terminology and Definition

AUIs share a close relationship with intelligent user interfaces (IUIs) (Koelle 2004). IUIs have been defined as follows:

> "*Intelligent user interfaces (IUI) are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture)*" (Maybury 1998).

The above definition suggests that IUIs are able to exhibit intelligence regarding system functionality and the preferences of the user and reflect this intelligence in the interface they present to the user. It is difficult to find a generally accepted distinction between IUIs and AUIs, with three differing views apparent in the existing literature:

- *AUI and IUI as synonyms:* The two terms are often used interchangeably and refer to similar types of systems (Alvarez-Cortes *et al.* 2007);

- *AUI as an IUI component:* AUIs have also been described as one possible component of IUIs (Koelle 2004). Intelligent help systems and intelligent tutoring systems have been identified as other IUI components (Dieterich *et al.* 1993); and

- *AUIs as a subclass of IUIs:* AUIs are often described as a subclass of IUIs (Ross 2000).

The third view, which regards AUIs as a subclass of IUIs, is the most dominant in AUI literature. IUI systems tend to place a greater emphasis on the input and output channels between the user and the system (Spenceley and Warren 1998; Ehlert 2003). Multimodal interaction and natural language interfaces (sometimes used for output, as well as input) are often a major focus of IUI systems, but are largely ignored in AUIs (Maybury 1998).

A distinction is sometimes drawn between adaptive systems and AUIs, with AUIs identified as one type of adaptive system (Krogsæter and Thomas 1994). For the purposes of this research, however, a broader definition will be adopted that draws no distinction between the two:

> "*An adaptive user interface is a software artefact that improves its ability to interact with a user by constructing a user model based on partial experience with that user*" (Langley 1999).

where a user model is defined as:

> "*…an explicit representation of the properties of an individual user; it can be used to reason about the needs, preferences and future behaviour of the user*" (Ross 2000).

A variety of other terms have also been used to refer to such systems in literature and commercially, including "User Modelling Systems", "Software Agents", "Intelligent Agents" and systems that provide "personalisation" (Jameson 2003).

3.3 Classification of AUIs

The most notable classification of AUIs in literature is the division between adaptable and adaptive systems (Opperman 1994; Fischer 2001). The term adaptable is used to refer to systems in which the user retains full control over adaptations to the system. Adaptive systems are those in which automated adaptations can be performed by the system, based on the system knowledge bases, which contain information the system has acquired regarding users and their tasks. In reality, the distinction is not always as clear as this, as many systems are actually hybrids of the two classes. To give a clearer picture of the different classes of AUIs, Dieterich *et al.* (1993) split the process of adaptation into four different phases (from the user's point of view). These four phases are:

- *Initiative*: Either the system or the user decides to perform an adaptation;

- *Proposal*: Alternatives for adaptation are proposed;
- *Decision*: One of the alternatives has to be chosen; and
- *Execution*: The adaptation is performed.

Each of these four phases can be performed or controlled by either the system or the user, leading to sixteen possible combinations, one of which is shown in Figure 3.1.

	System	User	
Initiative	✓		System initiates adaptation
Proposal	✓		System proposes some change/alternatives
Decision		✓	User decides upon action to be taken
Execution	✓		System executes user's choice

Figure 3.1: An example AUI classification showing division of control between the system and user (in this case, user-controlled self-adaptation (Dieterich *et al.* 1993)

Some common combinations have been assigned labels. On the one extreme is *Self-Adaptation* where the system is responsible for all four phases. At the other end of the scale is *Adaptation* where the user is in control of all phases of the adaptation process. One example of a hybrid is *Computer-Aided Adaptation*, where the user decides adaptation is needed, the system presents alternatives of which the user selects one, and then the system performs the adaptation as requested.

While placing control in the hands of the user has obvious advantages in terms of user control and freedom, it has been found that users often do not make use of available functionality to adapt the user interface (Opperman 1994). To solve this problem, systems have been developed which combine the features of adaptive and adaptable AUIs. An example of such a system is the *Flexcel* system (Krogsæter *et al.* 1994) which modifies Microsoft's spreadsheet application, *Excel.* Suggested adaptations are presented automatically by the system, but the user retains control and is able to ignore these suggestions. The adaptation process is also entirely transparent, with tutorials provided on how the adaptation works and how to use it effectively. The user is notified of new suggestions by a flashing light on a toolbar, rather than by an intrusive dialogue popping up which could be very frustrating if unwanted.

3.4 Benefits of AUIs

The users of any application are heterogeneous. Their learning abilities, psycho-motor skills, cognitive abilities, preferences, expectations, motives, requirements and experience levels may vary considerably (Browne *et al.* 1990; Alvarez-Cortes *et al.* 2007). The majority of software systems developed today provide a single, generic interface to every user. All users are also presented with the same options, regardless of their preferences and data is presented to all users in the same format. Some applications do provide minor customisation options regarding colour, font size and toolbars, but in essence the interface is static and unintelligent. The user is forced to adapt to the interface, rather than the other way round. Today's interactive systems are used by a wide variety of people, with varying skill-levels. In order to cater for as wide a user population as possible, designers often attempt to make an interface as simple as possible; in essence catering for the lowest common denominator amongst the user population (Kobsa 2004). If interfaces could adapt to the individual user, many potential benefits could be achieved. AUIs provide such a means of tailoring the system and its interface to the individual.

Browne *et al.* (1990) identified a number of high-level potential benefits of AUIs, including improving accuracy, efficiency and aiding in user learning. In the two decades since, the research area has matured through experimentation and a number of benefits have emerged. AUIs have been touted as potential solutions for problems such as information overload and filtering, learning to use complex systems and automated task completion (Höök 1999). AUIs have also been employed to provide personalised content recommendations and to tailor the presentation of information according to user preferences. It has also been suggested that AUIs also provide the potential to deliver the benefits of customised software at a lower development cost per user (Ross 2000). These benefits depend on the particular type of AUI and will be discussed together with the different types of AUIs in Section 3.9.

Despite many potential benefits, several problems have been encountered which have prevented the full potential of AUIs from being fulfilled. These problems and shortcomings will be discussed in the following section.

3.5 Problems and Shortcomings of AUIs

AUIs provide many potential benefits, but are not without problems. One of the fundamental usability heuristics identified by Nielsen (1994a) is "User Control and Freedom". AUIs can be seen as taking control away from users and putting them at the mercy of the system.

Other possible problems and pitfalls include the following (Opperman 1994; Jameson 2003; Paymans *et al.* 2004):

- *Privacy* issues related to the collection of user interaction data. This is particularly relevant if the data is of a confidential nature. Users are often reluctant to provide personal information, particularly on the Internet (Kobsa 2004).
- The user's mental model of the system can become *confused*. Changes in the interface can be very confusing to users as they may not understand what is causing the changes (Tsandilas and Schraefel 2004).
- *Learnability* can be negatively affected. Users like a consistent interface that they can learn and that allows them to become efficient over time. Adaptations can lead to a lack of predictability and frustration on the part of the user.
- *Obtrusiveness* can also be a problem, especially with regard to intelligent help systems and recommender systems. Constant pop-ups and annoying messages can irritate and frustrate users.

Various solutions have been proposed to the issue of loss of user control. Several AUI systems have been developed which put control back in the hands of users. Users have the power to disable the adaptive features of the system if they so desire. Other systems make an effort to make the adaptation process more transparent to users in an effort to aid their understanding and to help them develop a more accurate mental model of the functioning of the system (Kühme 1993). Jameson (2003) provides a detailed discussion of the usability challenges of AUIs, including possible preventative and compensatory measures.

3.6 AUI Components

The basic premise behind AUIs is the realisation that users are different and have different needs from a user interface (Korvemaker and Greiner 1999). Non-adaptive systems provide an interface to meet the needs of what designers view as the typical user. If the system is to adapt to different users, knowledge is required regarding each user. As per the definition in Section 3.2, this is done using a user modelling component which is built into the AUI system, modelling certain user characteristics and behaviour.

User models are probably the most crucial knowledge base in AUIs, but are not the only type of model used by such systems. Along with user models, task models are the other most significant model in AUIs (Krogsæter and Thomas 1994). Task models can be either static or dynamic, modelling either standard system tasks, or the task currently being performed by the user. However, knowledge about users and their tasks is not enough to facilitate adaptation. The AUI also needs to have knowledge of itself in order to know how it is able to adapt to the user. For this purpose, a variety of other models are sometimes also employed. These models include a domain model, a dialogue model and a system model (Dieterich *et al.* 1993; Krogsæter and Thomas 1994).

Opperman (1994) identifies adaptive systems (which according to the definition in Section 3.2 are AUIs) as consisting of three high-level components:

- *Afferential component*: Observes and records user behaviour and system reactions;
- *Inferential component*: Analyses the gathered data to draw conclusions and decides how the system should adapt; and
- *Efferential component*: Modifies how the system behaves. This could be one automatic effect, or the user could be presented with a range of alternatives.

Several general architectures for AUIs exist, three of which are shown in Figures 3.2 to 3.4. The first is a very high-level architecture (Figure 3.2), depicting AUIs as having adaptivity goals (Karagiannidis *et al.* 1995). These are achieved through adaptivity rules, which rely on knowledge stored in a variety of models, which are built through modelling user interaction. The adaptivity rules are used to make changes to the interface presented to the user.

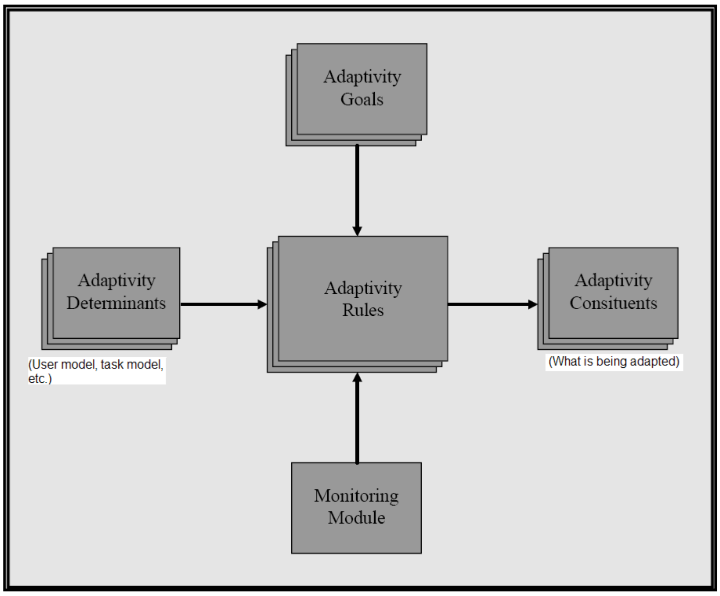

Figure 3.2: Attributes of an AUI. Modified from (Karagiannidis *et al.* 1995)

Kühme (1993) presented a more low-level, although slightly simplistic model of AUIs (Figure 3.3). He splits AUIs into four components, namely the monitor (which records information regarding user interaction), the user model (which stores this information, possibly after processing it using user modelling), the adaptor (which acts on the stored information to adapt the interface) and the adaptable dialogue (the interface).

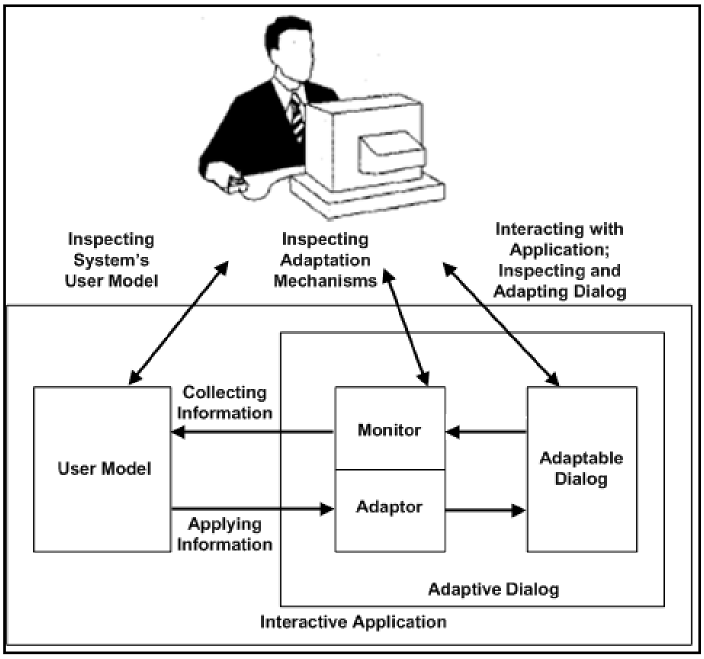

Figure 3.3: Kühme's model for AUIs (Kühme 1993)

Kules (2000) incorporates additional models into his model of an AUI (Section 3.4). He describes an Analysis Engine component, which is responsible for updating the user model and for performing adaptations to the interface.

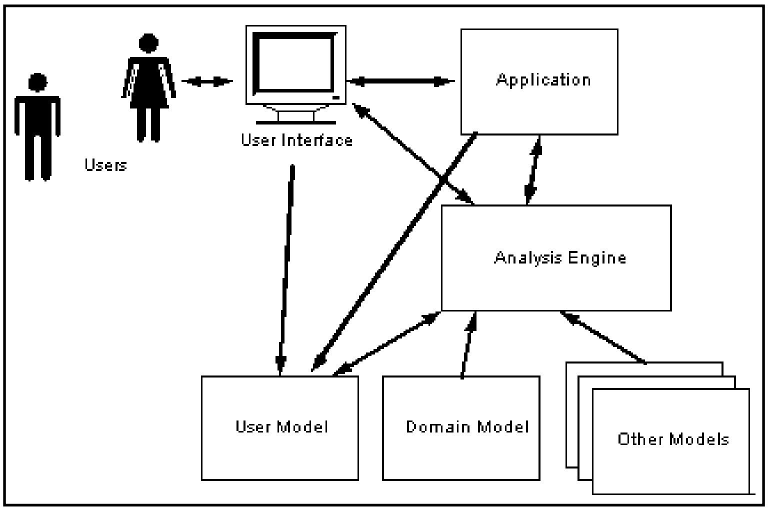

Figure 3.4: Kules' model for a typical adaptive system incorporating a user model (Kules 2000)

The above three models show us that while the basic components of AUIs identified in literature are similar, the terms used to describe these components may vary. The list below summarises the typical components of an AUI, drawing on the above models as well as the work of Opperman (1994):

- A *Monitoring Module* – observes and records user interaction;
- A *Knowledge Base* – which contains a variety of models and information needed by the AUI, including:
  - *User Model* – models user characteristics and behaviour;
  - *Task Model* – models user and/or system tasks;
  - *System Model* – models system functionality and capabilities; and
  - *Domain Model* – contains information about the application domain which is used by the other system components;
- *User Modelling Component* – dynamically updates the user model taking into account observed user behaviour;
- *Inference Engine* – decides when and how to adapt; and
- *Adaptor* – performs the adaptations.

The discussion of the different components will be organised into three sections relating to the afferential, inferential and efferential components which make up any AUI (Opperman 1994).

3.7 Afferential Component

The afferential component of an AUI is concerned with observing user interaction with the system in order to inform the adaptation process. This process can be split into two phases, namely data collection and data storage.

Adapting to the individual user requires collecting data about each user. This can be done either explicitly or implicitly. Explicit data collection is done by asking the user direct questions, usually by presenting them with a form to complete before using the system for the first time. This approach, while it ensures accurate information is collected (assuming the user answers truthfully), is not ideal. Requiring users to fill in forms before using a system is time-consuming and may be regarded as irritating by some users (Ross 2000). Some systems also request information that may cause users to have privacy concerns regarding the information they are asked to divulge (Jameson 2003). Examples of data typically required includes age, sex, location and preferences regarding content (de la Flor 2004). The information collected in user models of this type is typically static and only needs to be collected once.

Dynamic user models implicitly collect data by observing user behaviour. This method has the advantage of being unobtrusive, although designers need to be careful that the recording of data does not interfere with the performance of the system. Implicit data collection typically involves recording user interaction with the system, in the hope of deriving user preferences and frequently executed tasks. This data collection technique may raise privacy concerns for some users (Jameson 2003).

Some user models employ a combination of explicit and implicit data collection techniques. Others use a third category of data collection, which involves requiring the user to solve special tasks and observing them doing so (Domik and Gutkauf 1994). This approach can be used to extract information regarding the user's perception of colour, motor coordination and other cognitive abilities. Domik and Gutkauf (1994) discuss the use of educational games to discover user capabilities and characteristics.

The results of this data collection process are stored in the user model. The user model is often not the only model present in AUIs, with other possible models including tasks models, system models and domain models. These all play an important role in providing supporting knowledge to AUIs. These models are discussed in the sections which follow.

3.7.1 Task Model

Task models are defined as *"…logical representations of user activities"* (Lacaze and Palanque 2004). Task modelling can either be static or dynamic. Static task modelling involves modelling the different tasks that can be performed with the system. Dynamic task modelling attempts to identify the task in which the user is currently engaged (Krogsæter and Thomas 1994). Task models which model the user's current task are equivalent to very short-term user models (Ross 2000). Static task modelling is fairly well established as part of the user interface analysis and design process (Preece *et al.* 2007). Although recommended for use in the analysis of existing situations, rather than the design of new systems, task modelling has also been used in the design phase of systems (Eisenstein and Rich 2002;

Mitrovic *et al.* 2007). Task models are typically used in conjunction with user models to determine the user's task and goal. Some systems seek to identify the user's current task in order to offer automatic task completion (if this is possible) or assistance in completing the task (Krogsæter and Thomas 1994). Plan recognition also relies on task models and is often a crucial part of intelligent systems and tutoring systems, which seek to provide users with help and guidance relevant to what they are attempting to do (Jameson 2003).

3.7.2 System Model

In order for an interactive system to be adaptive, it must have some knowledge about itself. This is necessary in order to reason about how to adapt to the user. System models contain knowledge about the system, including the capabilities of the device, the functions available, the system architecture or the user interface (Reichenbacher 2004). The system model can also contain the adaptable parameters of the system and the current values of these parameters. If the user interface itself is to adapt (as is usually the case with an AUI), then the system model is sometimes referred to as a dialogue model (Krogsæter and Thomas 1994). It is important that this model is flexible, to allow for adaptations to the interface to be made (Dieterich *et al.* 1993).

3.7.3 Domain Model

The domain model contains knowledge about the real-world domain (Reichenbacher 2004). This knowledge serves as an information source for other components of the system. Domain models are typically fairly static in comparison with other models and contain knowledge describing the domain of the system in question (Krogsæter and Thomas 1994). One example of a domain model is that described by Kofod-Petersen and Aamodt (2003), in which the relationship between different entities in an airport domain is described, as well as how the context of these entities affects their relationships.

The terminology is far from standard, however, as can be seen from the description of the "system model" for a personalised tour guide given by Fink and Kobsa (2002):

> *"The System Model encompasses information about the application domain…"*

This definition is significantly similar to the description of a domain model given above, illustrating the inconsistencies that exist in the use of the different terms in previous work.

3.7.4 User Model

The user model is the most critical component of any AUI (as can be seen from the definition of an AUI given in Section 3.2). The process of user modelling contributes two components to the typical architecture of an AUI: the user model itself (typically stored in a database if used in a multi-user application), as well as a user modelling component (Krogsæter and Thomas 1994). User modelling is a diverse research area and can be classified in many different ways. The following four dimensions are among the most common (Ross 2000):

- *What is modelled*: The typical user, user groups or the individual user;
- *Source of modelling information*: Explicitly collected information or knowledge gained from analysing data collected from observing user interaction with the system;
- *Update methods*: A static or dynamic model; and
- *Time sensitivity of the model*: Short-term specific information or long-term generic information.

Each of the above dimensions has a significant effect on the form of the user model and the user modelling process. The first dimension, "What is modelled", has been approached in several different ways. Systems have historically used one of three different approaches, namely creating a single user model for all users; creating user models for different stereotype groups; or creating individual user models (Dieterich *et al.* 1993). Attempting to capture the characteristics of all users in a single model is essentially what designers of static, traditional software systems attempt to do during the design process. This approach does not involve catering to the individual user and consequently is of little interest to this research. The stereotyping approach involves classifying users as belonging to one of a set of possible user groups. This typically involves classifying users as novice, intermediate or expert and providing an interface that corresponds to these abilities (Crow and Smith 1993). Generally, however, AUIs attempt to adapt to the characteristics of the individual user.

The second and third dimensions of a user model (the source of the modelled information and the update methods) affect whether it is necessary to observe user interaction. If the user model consists purely of static, explicitly obtained information, then there is no need to monitor and record user behaviour. In this case, the user model is not updated and adaptations are performed based on the information which is gathered the first time the system is used. If a user model is based on user interaction data, the user model can also be updated during system use or only once, at system start-up or shutdown.

User models also vary according to the time sensitivity of the model. User models can have a very short-term memory. Alternatively, long periods of user interaction can be recorded and this data used by the user modelling algorithm to learn user preferences. The specific application domain affects the time period that is relevant to the user modelling process.

3.8 Inferential Component

User modelling consists of more than just collecting data about users. A modelling process is needed to turn this data into useful knowledge. User modelling techniques vary dramatically, from fairly simple systems which record the frequency with which different menu options are selected (Sears and Shneiderman 1994) to complex systems involving machine learning algorithms of different varieties (Langley 1999). The few IUIs that have succeeded commercially have used very simple adaptations based on simple knowledge of the user, rather than the complex user modelling architectures favoured by researchers (Höök 1999). This has led to research into minimalist user modelling techniques, which attempt to do “just enough” to make worthwhile adaptations to the system in question (Strachan *et al.* 1997). The need to improve the value / effort ratio of user modelling techniques has been identified as one of the future challenges of user modelling (Fischer 2001). There is little point in spending months implementing a complex user modelling component if the benefits for the user are minimal. Nevertheless, some very effective, complex user modelling techniques have been developed. Although our focus is on MMV systems, which run on resource-constrained devices, the use of distributed architectures is possible. User modelling can be performed on a powerful server and only the results stored on the mobile device. Hybrid approaches have also been proposed (Zhang *et al.* 2006), where user models are stored at differing levels of detail on the mobile device and server (Papadogiorgaki *et al.* 2007).

Two main approaches to user modelling exist (Zukerman and Albrecht 2001):

- *Content-based Learning:* Adaptations are performed based on previously observed user behaviour. This technique has the disadvantage that some initial training data is required from the user before useful predictions can be provided.
- *Collaborative Learning:* Adaptations are made based on previously observed behaviour of similar users (similarity is determined based on observed interaction or

demographic profiles). A model is built based on a group of users and then applied to an individual user. This technique potentially allows useful predictions or recommendations to be made to users from the first time they use the system, as other users can be used to provide training data for the algorithm. Collaborative filtering (in which collaborative learning is applied to filter the information selection or provide recommendations) is one of the few techniques that have been successfully used commercially (Höök 1999). E-commerce retailers can use collaborative filtering to make informed recommendations to the user (de la Flor 2004). This technique has also been used in combination with other techniques successfully in the past (Jameson 2003). Collaborative filtering has thus proved a popular technique in AUIs (Hedberg 1998; Hui and Boutilier 2006).

The discussion of user modelling techniques which follows will consider simpler techniques first, followed by more complex methods. There is some overlap between the categories identified and many hybrid techniques have been proposed which combine user modelling techniques from different categories. Both content-based and collaborative techniques have been applied in many of these categories.

3.8.1 Simple User Modelling Techniques

Many of the early AUIs that were developed involved optimising menu structures (Greenberg and Witten 1985; Mitchell and Shneiderman 1989; Sears and Shneiderman 1994). These systems relied on simple user modelling processes that recorded how often different menu options were selected and, in the case of *Split Menus* (Sears and Shneiderman 1994), promoted the most frequently selected items to the top of the menu. More recently, Findlater and McGrenere (2008) used a combination of the most recently and frequently selected items to adapt the ordering of menu options (Figure 3.5).

1. Set top section to the *most recently* selected item and the *two most frequently* selected items (as pre-calculated from the selection stream)
2. If there is overlap among these three slots or if this is the first selection in the stream (i.e., no recently selected item exists)

    Then the third most frequently selected item is included so that 3 unique items appear in the top section

3. Order top items in the same relative order as they appear in the bottom section of the menu

Figure 3.5. Base adaptive algorithm for menu ordering (Findlater and McGrenere 2008)

Strachan *et al.* (2000) proposed another fairly simple approach to user modelling. They identified the complexity of most of the user modelling techniques developed in research systems as a major obstacle to the adoption of these techniques in commercial systems. They developed a simple user model which uses basic information, such as job titles, expertise and system usage information, to identify the user as a novice, intermediate or expert user stereotype (Figure 3.6). Their user modelling system is designed to provide maximum assistance to novice users, while minimising interruption for expert users. Assistance and interference from the AUI decreases in inverse proportion to the user's skill level.

User Name:	Peter Planner
Job Title:	Company Representative
Frequency of Use:	11 Days
Date of Last Access:	November 3, 1999
Last Update:	September 30, 1999
User Type:	NFP
Financial Planning Expertise:	Low
TIMS Experience:	2
Windows Experience:	1
User Complexity Rating:	2
TIMS Strategies: name: *	RRSP Maximizer
counter:	6
date of last use:	September 30, 1999
demo counter:	2
date of last demo:	August 20, 1999
demo refusals:	0
Assistant Modules: name: **	Planning Assistant
counter:	8
date of last use:	November 3, 1999
demo counter:	1
date of last demo:	August 20, 1999
demo refusals:	0
User Settings:	
Show Task Demonstrations	False
Show Strategy Interactions	True
Suggest Strategies	True
Data Entry Assistance	True
*For each individual strategy	
**For each of the Assistants	

Figure 3.6: Example of a minimalist user model (Strachan *et al.* 2000)

3.8.2 Classification Techniques

Many AUIs make use of machine learning techniques and specifically classification learning to perform user modelling. This set of methods includes neural networks, decision trees, case-based reasoning and probabilistic classifiers (Jameson 2003). Classification learning techniques usually employ supervised learning (Engelbrecht 2002), where training examples are provided along with correct classifications to allow the algorithm to learn the mapping between the input and its correct classification. Exact classification is not always possible, so alternatives are often presented to the user (Segal and Kephart 1999). This type of modelling technique is particularly suited to situations where textual information is available to identify items of a similar nature, for example movies, books, news articles or emails. A major disadvantage of this set of techniques is the requirement of training data. This can be handled in different ways. Users can either be asked to classify several test cases the first time they use the system (which they may or may not be willing to do), or the system can start off using a generic user model and then modify the model as it learns from observing the user. A wide variety of user modelling techniques make use of the classification approach.

Linear models attempt to predict the value of a dependent variable as a weighted sum of one or more independent variables (Wackerly *et al.* 2002). The method of least squares, which minimises the total squared error between predicted and observed values, is used to learn the optimal values for the various weights. Training data is necessary to facilitate the learning process.

Case-based reasoning (CBR) relies on a database of solutions to solve problems. Given a new case to classify, the CBR classifier consults its database of known solutions in order to see if a solution to that exact case is already stored. If not, solutions to similar cases are retrieved, and the CBR classifier attempts to merge these solutions into a solution for the new case (Han and Kamber 2006).

Naïve Bayesian classifiers use the values of several discrete-valued input attributes $\{A_1, A_2, \ldots, A_k\}$ to predict the value of another discrete-valued variable (C). They rely on the assumption that the values of the attributes used for prediction are independent of one another. Despite the fact that this assumption is frequently violated, Naïve Bayesian classifiers have been shown to deliver high levels of accuracy, with the added advantage of being computationally inexpensive (Elkan 1997).

Rule induction techniques attempt to learn a set of rules which classify observations based on the values of certain attributes. Training data is needed which provides observations and the corresponding correct classification. In order to extract rules from training data, a variety of sequential covering algorithms has been developed. These algorithms generally work by learning rules one by one. After each rule is learned, observations which match this rule are removed. Rules can be learned in a variety of ways. One method is to greedily add conditions to the left side of an IF…THEN rule, such that the attribute and corresponding value selected provides the maximum improvement in the quality of the rule. Quality can be measured in a variety of ways, often using the accuracy of the rule (high accuracy implies the rule correctly classifies observations with a high probability) (Han and Kamber 2006). Rules can be represented as IF…THEN statements, decision trees or conditional probabilities (Zukerman and Albrecht 2001).

Neural networks and *genetic algorithms* are among a range of techniques from the field of computational intelligence that can be used to perform user modelling. Neural networks attempt to model multi-dimensional decision surfaces by expressing these surfaces using layers of nodes and then learning the weights between these nodes in order to minimise classification error. Genetic algorithms rely on the principles of evolution, evolving an initial starting population of candidate solutions through cross-over (in which solutions are combined) and mutation (in which random changes are made to attributes) until the classification error is sufficiently small (Engelbrecht 2002).

3.8.3 Clustering Techniques

Clustering techniques, in contrast to classification techniques, typically use an unsupervised learning approach, where the correct classification of training data is not available. An n-dimensional space is divided into clusters in which the individual elements are similar to one another and different from the members of other clusters (Zukerman and Albrecht 2001; Engelbrecht 2002). These techniques can be useful for identifying related items where training data is unavailable.

3.8.4 Decision-Theoretic Models

The above user modelling techniques are largely data-driven and learn from observed interaction data. Another class of user modelling techniques exists that incorporates knowledge regarding users, their goals and the system with which they are interacting (Jameson 2003).

Bayesian networks are used to make decisions about the user's current state, need for help and likely next action (Jameson 2003). These networks are represented visually as directed acyclic graphs (DAGs), where each node represents a state (for example what the user is currently doing) and arrows between the nodes indicate probabilistic relationships, each of which is assigned a weight or probability. These relationships can usually be interpreted as causal (Jameson 2003), meaning that the state into which the arrows are directed has occurred as a result of the states from which the arrows originated. In a dynamic Bayesian network the probabilities are updated based on previously observed user behaviour. Bayesian networks typically learn using algorithms such as gradient descent, where training data is used to arrive at optimal values for the weights, by moving down the gradient of the error space in order to find the weight values which provide the most accurate classifications (Han and Kamber 2006).

Hidden Markov Models (HMMs) can be considered as the simplest case of dynamic Bayesian networks. HMMs are based on the Markov assumption, that the probability of an event occurring is based only on a fixed number of preceding events (in the simplest case, one). HMMs allow us to determine several hidden parameters, based on the values of observable parameters. In terms of AUIs, HMMs allow the most likely user action to be predicted, given the user's preceding actions. This is determined using the probability distribution of events which have previously followed the same set of preceding actions (Zukerman and Albrecht 2001). This is potentially useful in identifying user plans or goals (Han and Kamber 2006).

Stereotyping (Section 3.7.4) divides users into pre-defined groups (e.g. novice, intermediate or expert). Rules are built into the system in order to classify a user as belonging to a certain stereotype (Jameson 2003). Once a user is classified as belonging to one of these stereotypes, appropriate adaptations could be performed. For example, a novice user may require a simpler interface or additional assistance in comparison to an expert user.

3.9 Efferential Component

According to our definition, AUIs improve the system's interaction with the user. In order to do this, AUIs must act in some way on the knowledge stored in the various models in the Knowledge Base. Two aspects of the adaptation process need to be decided, namely when to adapt and how to adapt. Dieterich *et al.* (1993) distinguish three different times when adaptation can occur, namely before, during or between usage sessions. If the physical appearance of the interface is changing, care has to be taken not to do this more often than is necessary, as this could cause confusion for the user. The question of when to adapt is intrinsically linked to how the interface adapts. AUIs can use several different strategies to improve interaction with the user. Figure 3.7 shows a high-level view of the typical adaptation process of AUIs. Starting with the user, it shows how user input is processed to create knowledge which is stored in the different models. These models are then used in the adaptation decision making process. This leads to adaptations being applied which affect the output presented to the users or the interface with which they interact.

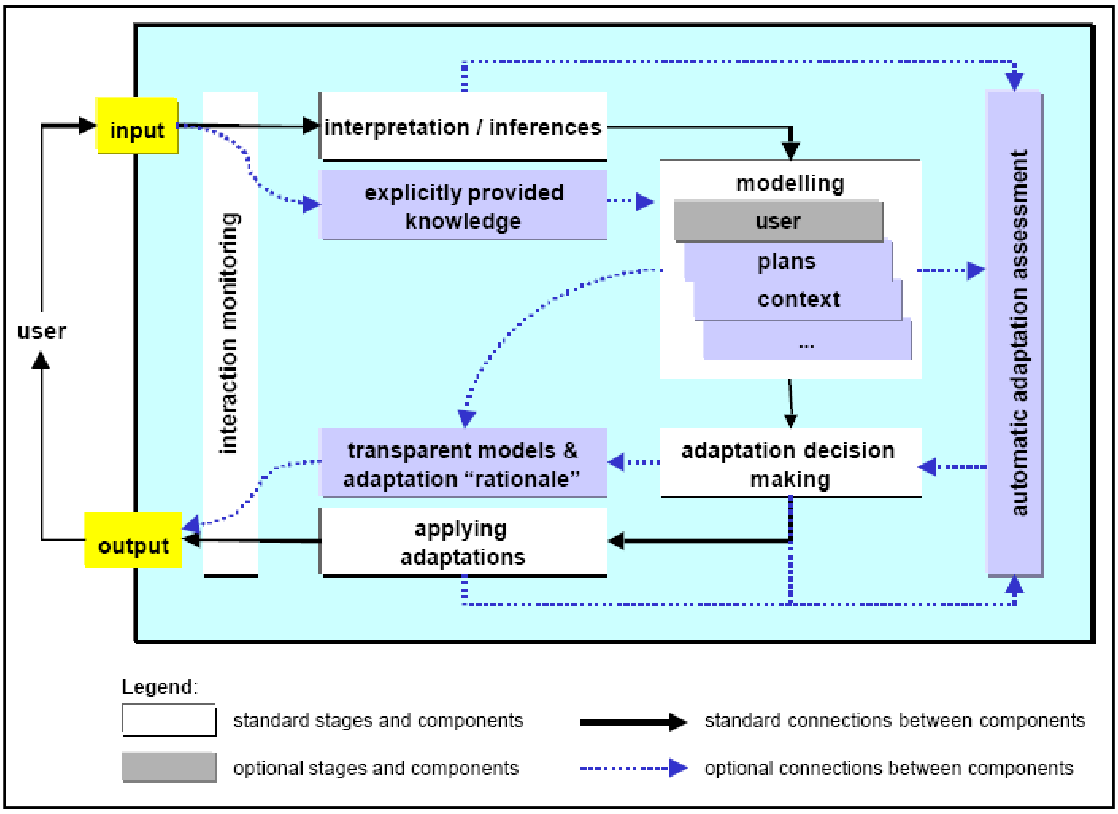

Figure 3.7. High-level model of adaptation in AUIs (Paramythis *et al.* 2001)

Some architectures specifically include an Adaptor component which performs adaptation (Figure 3.3), while in other architectures, this component is more implicitly implemented. This component is, however, a fundamental part of every AUI.

Jameson (2003) identifies nine useful functions of AUIs, split into two broad categories: those that support system use and those that support acquisition of information. The discussion that follows regarding the different classes of AUIs will include these nine functions, grouped into five categories, based on the techniques that have been used to achieve the different functions. Where possible, examples that are relevant to mobile computing and mobile visualisation will be highlighted, as these are the most relevant to this research. The five categories of AUIs to be discussed are:

- Interface adaptation;
- Task assistance and adaptive help;
- Recommender systems and information filtering;
- Information presentation; and
- Adaptive learning.

3.9.1 Interface Adaptation

This category of AUIs refers to systems that make physical adaptations to the user interface to suit the individual user (Jameson 2003). Many early AUIs provided for fairly fundamental changes to be made. Different interface variants were stored and based on the characteristics of the user, one of these was chosen and presented to the user (Totterdell and Rautenbach 1990). Changes could be as fundamental as replacing a direct manipulation GUI interface with a command line interface (Lennard and Parkes 1995). More recent examples of this class of AUIs have been more subtle. Various versions of Microsoft Office have made use of adaptive menus and toolbars. Only frequently used options are displayed, with the infrequently used options hidden. The rationale behind this is that users are able to quickly access frequently used options without having to navigate through a multitude of irrelevant options. Microsoft Research has experimented with a Split Interface (in which an extra toolbar containing frequently used options is added) and a Moving Interface (in which recently used options are promoted to the main toolbar) (Gajos *et al.* 2006). Another common interface adaptation technique (particularly in web systems) is adaptive navigation, where the navigation structure is modified to accommodate the individual user (Vasilyeva *et al.* 2005).

Interface adaptation has proved popular in mobile systems. Boström *et al.* (2008b) adapted the interface of their system for browsing mobile widgets (called Capricorn) by "greying out" infrequently used widgets and providing shortcuts for easy removal (Figure 3.8).

Figure 3.8: Interface adaptation in Capricorn (Boström *et al.* 2008b)

Bridle and McCreath (2006) compared several techniques for providing shortcuts to help users complete common tasks they perform with their mobile phones more efficiently, focusing specifically on outgoing communication tasks. The user was provided with a shortcut to perform an outgoing communication task (e.g. a shortcut to send a text message to a particular contact). They compared several fairly simple approaches, including decision trees (Section 3.8.2), Naïve Bayesian classifiers (Section 3.8.2) and suggesting the most frequently and recently performed outgoing communication task (Section 3.8.1). Their results showed that using these relatively simple techniques, considerable interaction effort could be saved on the part of the user.

Findlater and McGrenere (2008) used similar simple algorithms to adapt menus in a simulated small screen setting. Using the base adaptive algorithm shown in Figure 3.5, they showed that significant benefits could be achieved by adapting the ordering of menu options on small screen devices. They suggest that the benefits of interface adaptations of this kind could even be greater for mobile devices than for desktop systems.

Interface adaptation has been applied in MMV systems. Figure 3.9 shows an example of an MMV system which supports users searching for points of interest (POIs). Various controls are provided to search for POIs matching certain criteria, organised into tabbed controls. The tabs are ordered based on previously observed user behaviour (Carmo *et al.* 2008).

Figure 3.9: An example of interface adaptation in an MMV system. Tabs are ordered based on previous user behaviour (Carmo *et al.* 2008)

3.9.2 Task Assistance and Adaptive Help

Knowledge stored in user and task models allows AUIs to make inferences about the user's preferences with respect to certain tasks. Several systems have been developed to take advantage of this knowledge by providing the user with assistance and advice in completing routine tasks. Several of these systems are referred to as "Plan Recognition" systems, which aim to infer the user's future actions and then attempt to respond accordingly (Zukerman and Albrecht 2001). In many of these cases, the system responds by offering to complete or assist in completing the user's current task. MailCat is an AUI that assists users in filing incoming emails (Segal and Kephart 1999). The system attempts to learn the folder in which a new message is likely to be filed by observing user behaviour. Three buttons are provided, each suggesting a likely destination folder. If the user recognises one as being correct, the corresponding button can be clicked to file the message. Korvemaker and Greiner (1999) describe a UNIX command shell which attempts to predict the next command users will enter based on their past behaviour. If the prediction is correct, users can hit the 'Enter' key, or otherwise type in a command of their own.

Chen and Chen (2007) presented an approach which, similar to MailCat, provides assistance with a sorting task. Both their algorithm and the MailCat algorithm can be described as classification algorithms (Section 3.8.2). Chen and Chen's algorithm focuses on helping users sort photos captured using a mobile phone into collections which correspond to events. The algorithm attempts to match users' photo taking behaviour to a Poisson statistical process, therefore expecting a fairly predictable interval between photos that belong to the same event. Photos outside of this interval could belong to a new event. When the algorithm is unsure whether a new item belongs to a new collection or the current collection, location information is incorporated to assist in the classification process.

Systems of this kind are often referred to as adaptive help systems, or Intelligent Help Systems (IHS) (Dieterich *et al.* 1993). Adaptive help systems attempt to adapt to the user in terms of how and when to provide help. They are generally divided into two main classes, namely active and passive adaptive help systems. Passive help is only provided when requested by the user. Active help, on the other hand, can be displayed at any time when the system feels that the benefit of the provided help outweighs the cost of providing it, both in terms of effort and the possible resulting irritation to the user (Brusilovsky and Schwarz 1997). Bayesian Networks (Section 3.8.4) have been used to create probabilistic models to help decide when and how to provide help to the user, based on factors such as the user's independence and tendency to be distracted (Hui and Boutilier 2006). In general, adaptive help systems attempt to identify when the user is making errors, or performing tasks in a sub-optimal manner. One of the biggest challenges is to decide when or if to interrupt the user (Brusilovsky and Schwarz 1997).

MMV systems have also been adapted to provide users with assistance relevant to their current task (Chalmers *et al.* 2001). Context information can be harnessed in several ways to support user tasks, such as visualising the user's current location when performing navigation tasks, or visualising the opening hours of restaurants when the user is searching for somewhere to eat (Von Hunolstein and Zipf 2003)

3.9.3 Recommender Systems and Information Filtering

Recommender systems are a very common class of AUI. Such systems are particularly popular in E-commerce in order to recommend products that users might be interested in, given the behaviour that they have displayed in the past. Recommendations are either based on the individual user's past behaviour (content-based) or on the behaviour of similar individuals (collaborative techniques, Section 3.8). Typically, systems of this type are used to recommend products, movies or news articles in which the user would be interested (Jameson 2003), but other more creative uses of this type of adaptation have also been implemented. Recommender systems have been developed to provide personalised route recommendations and stock trading advice (Rogers *et al.* 1999; Yoo *et al.* 2003). User feedback is used to improve the accuracy of future recommendations. Such systems have proved successful commercially (Höök 1999) and are fairly easy to implement, with a readily available source of training data and feedback.

The Capricorn system (Section 3.9.1) represents a mobile example of the use of recommendation and information filtering. In addition to the interface adaptation described in Section 3.9.1, Capricorn also provides collaborative information filtering and content recommendations using an association rule algorithm (Boström *et al.* 2008b). Content is also adapted according to the user's context. Mobile recommender systems have also been developed to deliver personalised restaurant recommendations, allowing for user feedback to improve system recommendations (Ricci and Nguyen 2007). Figure 3.10 shows a screenshot of a mobile system which allows users to respond to system recommendations, in this case allowing them to refine search parameters. Personalised tourist attraction recommendations are another domain that has been targeted (Ardissono *et al.* 2003).

Figure 3.10: A mobile restaurant recommendation system, allowing the user to provide feedback regarding recommendations (Ricci and Nguyen 2007)

An example of an MMV system applying this class of adaptation is Gulliver's Genie (O'Grady and O'Hare 2004). Gulliver's Genie is a personalised mobile tourist guide that uses user modelling to adapt to the user's context and preferences. Issues of switching output modalities and map rotation are also addressed and personalised to the user. Gulliver's Genie is not the only system of this nature to be developed, as several personalised tour guide systems have been developed, delivering tourist-related information that matches users' preferences and behaviour (Fink and Kobsa 2002).

3.9.4 Information Presentation

The previous section dealt with recommending information to users that they may be interested in. AUIs can extend beyond selecting *what* information to present to the user and can also adapt *how* that information is presented. Browne *et al.* (1990) emphasise the differences in users' cognitive strategies and abilities. Such factors can have a dramatic effect on how effective information presentation is for an individual user. One presentation of information may be suited to one user, but inappropriate for another. Users' productivity can be greatly improved if information is presented in a format which they prefer and can easily understand and interpret. Domik and Gutkauf (1994) advocate adapting visualisations to, amongst others, the desires, disabilities and abilities of the user. The volume of information presented to the user, the perspective from which that information is viewed and the format in which the information is presented have all been identified as important aspects of information presentation to be adapted (Koelle 2004).

Several MMV systems have implemented adaptations of this class. Figure 3.11 shows an example from an MMV system for assisting users in performing navigation tasks. The presentation of the suggested route is adapted depending on whether the user is walking, or taking a bus (in which case the bus stops are emphasised on the map) (Reichenbacher 2004).

Figure 3.11: Presentation adaptation in an MMV system. The visualisation of a planned route is adapted according to the whether the user is walking (left) or taking a bus (right) (Reichenbacher 2004)

3.9.5 Adaptive Learning

AUIs which fall into the adaptive learning category are often referred to as Intelligent Tutoring Systems (ITS) (Hefley and Murray 1993) or Learner Modelling Systems (LMS) (Jameson 2003). These systems attempt to adapt to the user's knowledge of the domain in question, learning style and processing of the current problem or concept (Jameson 2003). Adaptive learning systems are often based on the principle of incremental learning, where the user is first introduced to basic concepts, before moving on to complex concepts which require understanding of the basics (Brusilovsky and Schwarz 1997). The subject matter in question is presented in the order which the system (based on analysis of the user model) believes to be the optimal order for that user to follow. A less common application of adaptive learning systems is incremental interfaces, where the user learns to use a complex interface by gradually being introduced to more complex functionality once they have mastered functionality at the previous (lower) level of complexity (Brusilovsky and Schwarz

1997). This class of adaptations is more suited to desktop systems, as mobile systems to facilitate learning are still rare.

In the next section, the use of AUIs in MMV systems is addressed. Existing research is analysed in order to determine the variables which are used as inputs to the adaptation process and which of the above classes of AUIs have been implemented in MMV systems.

3.10 AUIs for MMV

AUIs provide the potential to overcome many of the limitations of mobile devices (Al-bar and Wakeman 2001; Chittaro 2004). Adapting the user interface has been identified as one of the most important aspects to be considered when designing mobile visualisation systems (Karstens *et al.* 2003). Any visualisation system which takes into account the needs and abilities of its users is likely to be an improvement on non-adaptive systems which present every user with the same information in the same format (Domik and Gutkauf 1994). Researchers have long realised that merely scaling information down for mobile devices is insufficient and that there is a need for AUIs and personalisation in a mobile environment (Billsus *et al.* 2002). It has been suggested that much of the existing research into AUIs should be revisited in a mobile context and that some approaches that have failed on the desktop may prove more successful on small screen devices (Findlater and McGrenere 2008).

For the purposes of this research, the focus is on AUIs for MMV. The problems associated with mobile visualisation are often aggravated due to the nature of MMV and additional problems unique to this class of systems exist (Section 2.5.3). The need for adaptation in MMV systems has previously been recognised and approached in a variety of ways by different researchers. Existing approaches differ considerably in the variables that drive adaptation and in the elements of MMV systems that are adapted. In the following two sections, existing research in terms of the use of AUIs in MMV will be categorised according to the variables used as input to the adaptation process (Section 3.10.1) and the resulting adaptation effects (Section 3.10.2).

3.10.1 Adaptation Input Variables

MMV systems can adapt to several different input variables. Looije *et al.* (2007) proposed adapting MMV systems according to four different elements:

- *User:* The system should adapt to the user's preferences, knowledge and skills (Section 3.7.4).
- *Task:* The system should adapt to the task the user is performing and to external tasks they may be performing while using the MMV system (e.g. walking) (Section 3.7.1).
- *System:* The system should adapt to differing device capabilities and variables such as battery life and network connectivity (availability and signal strength) (Section 3.7.2)
- *Environment/Context:* The system should adapt to the user's environment, including location, lighting and noise. Other elements of context include the user's current activity and proximity to other objects or people.

The first two variables, namely the users and their tasks, are essential parts of any AUI (Section 3.6). The last two variables, namely the system and context, are important because of the nature of mobile systems. Mobile devices differ widely in many respects, including screen size, screen resolution, input mechanisms, hardware and operating systems. In order to run on a wide variety of devices, MMV systems need to adapt to the capabilities of these devices. Mobile devices also provide us with the capability to sense the user's context, which provides a number of benefits and possibilities for adaptation (Section 2.4.1). Existing research into the adaptation of MMV systems can be classified according to the four input variables identified above.

User behaviour, capabilities and preferences have all been used as inputs in the adaptation of MMV systems (Wang *et al.* 2001; Zipf 2002; Fresco *et al.* 2003; Hampe and Paelke 2005). User tasks have also been used to adapt MMV systems, including mobile tourist guides (Chalmers *et al.* 2001; Von Hunolstein and Zipf 2003). Context is frequently incorporated in the adaptation of MMV systems, with location proving a popular input variable (Zipf and Richter 2002; Reichenbacher 2004). Finally adaptations to the capabilities of the user's mobile device have also been implemented on several occasions, in some instances also considering wider issues such as network quality of service (Chalmers *et al.* 2001; Fresco *et al.* 2003; Reichenbacher 2004).

3.10.2 Adaptation Effects

Just as adaptive MMV systems adapt to several different input variables, they also act upon these variables in several different ways. In Section 2.2, the Visualisation Reference Model (Figure 2.1) was discussed. This model identified three areas where human interaction plays a part in the visualisation process. These three areas, namely data transformations, visual mappings and view transformations, are also the areas were most of the adaptation effects discussed below take place. User interaction with these three areas of the visualisation process enables the system to learn the user's preferences and adapt accordingly. In response to changes in the input variables identified in the previous section, a wide variety of adaptations are possible which address the problems and shortcomings of MMV. These adaptations can be classified into the following four major types (Reichenbacher 2004):

- *Information Adaptation:* Adapting which information is visualised and how that information is structured (Section 3.9.3);
- *Visualisation Adaptation:* Adapting the presentation of information, including a wide range of parameters, from the level of detail to the visualisation format (Section 3.9.4);
- *Interface Adaptation:* Adapting the user interface through which the user interacts with the MMV system (Section 3.9.1).
- *Device Adaptation:* Adapting all aspects of the system to match the capabilities of the device, including screen size and interaction mechanisms. Device adaptation is not considered in the scope of this research (Section 1.4).

All four of the above forms of adaptation have been successfully employed in MMV systems. Information adaptation has been implemented to adapt the amount, classification and level of detail of information in MMV systems (Chalmers *et al.* 2001; Fresco *et al.* 2003; Looije *et al.* 2007). Visualisation adaptation has been used to adapt the orientation, layout, section, scale, generalisation and graphical elements of map-based visualisations (Harrie *et al.* 2002; O'Grady and O'Hare 2004; Reichenbacher 2004; Hampe and Paelke 2005). Adaptations of this type have ranged from fairly simple adaptations of the map symbols used (Nivala and Sarjakoski 2005), to advanced techniques such as Focus Maps (Section 2.6.4.2), which ensure that the maps presented match the user's requirements (Zipf and Richter 2002). Interface adaptations are comparatively rare, but have been implemented successfully in MMV systems (Carmo *et al.* 2008). Device adaptations are generally more low-level and ensure that the MMV system is able to run on different mobile devices (Chalmers *et al.* 2001; Fresco *et al.* 2003).

Reichenbacher (2004) also provided a lower-level view of the types of adaptation that can be performed within each of the three areas of adaptation which are within the scope of this research (Table 3.1). He identifies several forms of adaptation as being feasible for MMV systems, including filtering of unimportant information to reduce clutter, exchanging equivalent presentation formats (e.g. road map to topographical map) and adapting the symbols and colours used.

	Category		
	Information	**User Interface**	**Visualisation**

Element to be Adapted	• Encoding
	• Amount
	• Classification
	• Grouping
	• Level of Detail
	• Geographic Area
		• Functions
		  ○ Availability
		  ○ Granularity
		• Interaction
		  ○ Mode
		  ○ Style
			• Map Layout
			• Map Section
			• Map Generalisation
			• Method
			• Dimension
			• Graphical Elements
			• Typographical Elements

Table 3.1: Possible adaptable elements of an MMV system (Reichenbacher 2004)

One of the objectives of this research is to develop a model for the design of adaptive MMV systems. Before such a model can be proposed, existing models need to be reviewed. Several existing models, frameworks and architectures which are relevant to this research are discussed in the following sections.

3.11 Models, Frameworks and Architectures for AUIs

In order to design a model for adaptive MMV, existing models, frameworks and architectures were examined to determine how well these support the requirements of MMV and the input variables and adaptation effects identified in Section 3.10. Based on the focus of this research, only models, architectures and frameworks satisfying at least two of the following three criteria were considered as a basis for the model to be proposed in Chapter 4:

- *Mobility:* The model, framework or architecture should be relevant to the mobile computing domain;
- *Map-based visualisation:* The model, framework or architecture should incorporate the map-based visualisation process; and
- *AUIs:* The model, framework or architecture should incorporate an AUI into its design.

The models, frameworks and architectures discussed below will be analysed and compared in terms of their support for the adaptation input variables and effects identified in Section 3.10.

3.11.1 UbiquiTO Architecture

The UbiquiTO system (Cena *et al.* 2006) is an adaptive mobile tourist guide system and thus satisfies all three of the criteria for selection described above. The UbiquiTO system is designed to adapt to three of the four input variables identified in Section 3.10.1 (only the task variable is missing):

- *User:* User behaviour and preferences are modelled to tailor the system accordingly.
- *Device:* The system is adapted to suit the capabilities of the device on which it is running.
- *Context:* Primarily the user's location, but the date and time are also used to adapt the system.

These three variables are used to adapt two different aspects of the system. Firstly, the content being provided to the user can be adapted (Section 3.9.3). This can be in terms of the amount of information delivered, the type of information and the features provided with each recommendation. Secondly, the presentation and layout of this information is adapted (Section 3.9.4). The UbiquiTO architecture contains three different modules, one for each of the different variables which are used as input to the adaptation process (user model, device and context modules). A Watcher component observes and records user interaction in order to model user behaviour. Two components drive the adaptation. The first is the Recommender, which uses personalisation rules to select content which matches the user's preferences and context. The second is the Presentation Adapter, which adapts the amount of information to be displayed and its layout according to the user's preferences, context and device characteristics. The Interaction Manager manages interaction between the system and user, generating an XML object describing the personalised content for that user. The UI Generator then applies an XSL stylesheet to this XML object to format it as a web page to present to the user. The architecture of the UbiquiTO system is shown in Figure 3.12.

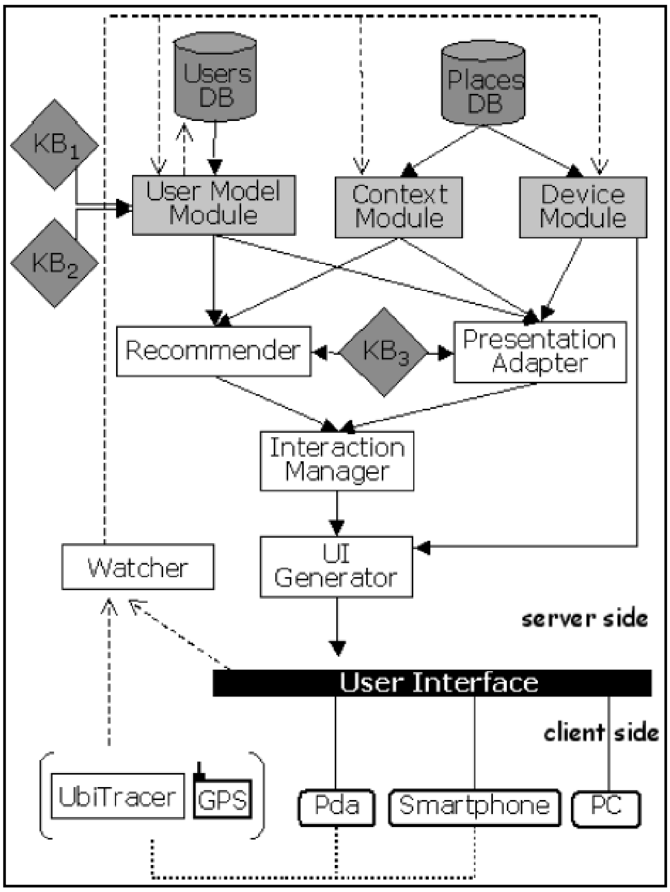

Figure 3.12: The architecture of the UbiquiTO Mobile Tourist Guide (Cena *et al.* 2006)

3.11.2 Adaptive Geovisualisation Model

Wang *et al.* (2001) proposed a high-level model for the architecture of an adaptive geovisualisation system (Figure 3.13). This model is an abstract model which describes adaptive geovisualisation in general and not MMV. It therefore satisfies two of the three criteria for selection (it focuses on map-based visualisation and AUIs). It emphasises the need to structure the visualised data in order to facilitate efficient querying (Section 3.9.3). This is also linked to the user modelling process, in order to ensure that personalised visualisations can be efficiently generated. The adaptive geovisualisation model is built around the user modelling process, using both explicit and implicit user modelling in order to adapt both the user interface (Section 3.9.1) and the map view presented to the user (Section 3.9.4). The Adaptive Geovisualisation Model is conceptual in nature, abstracting over low-level implementation details. The model that is described in the next section examines and describes the adaptation of MMV systems in more detail.

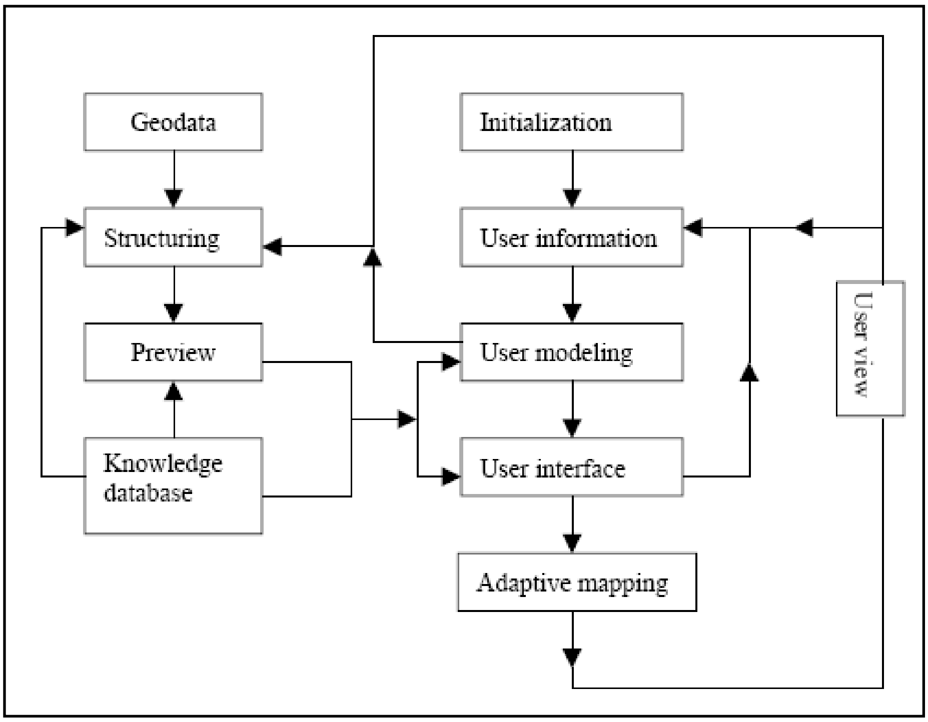

Figure 3.13: Adaptive geovisualisation system (Wang *et al.* 2001)

3.11.3 Mobile Cartographic Framework

Reichenbacher (2004) proposed the Mobile Cartographic Framework, which is of particular interest for this research, because it focuses specifically on adaptive MMV. It therefore satisfies all three criteria for selection identified at the beginning of Section 3.11. Figure 3.14 shows a high-level view of the framework, with the well-known Model-View-Controller design pattern as its backbone.

The adaptation process is divided into three components, namely adaptation objects (which are adapted), adaptation targets (adaptation objects are adapted to adaptation targets) and the adaptors (which perform the adaptation, transforming adaptation objects to targets). The three areas that can be adapted are the information content (Section 3.9.3), the user interface (Section 3.9.1) and the visualisation (Section 3.9.4).

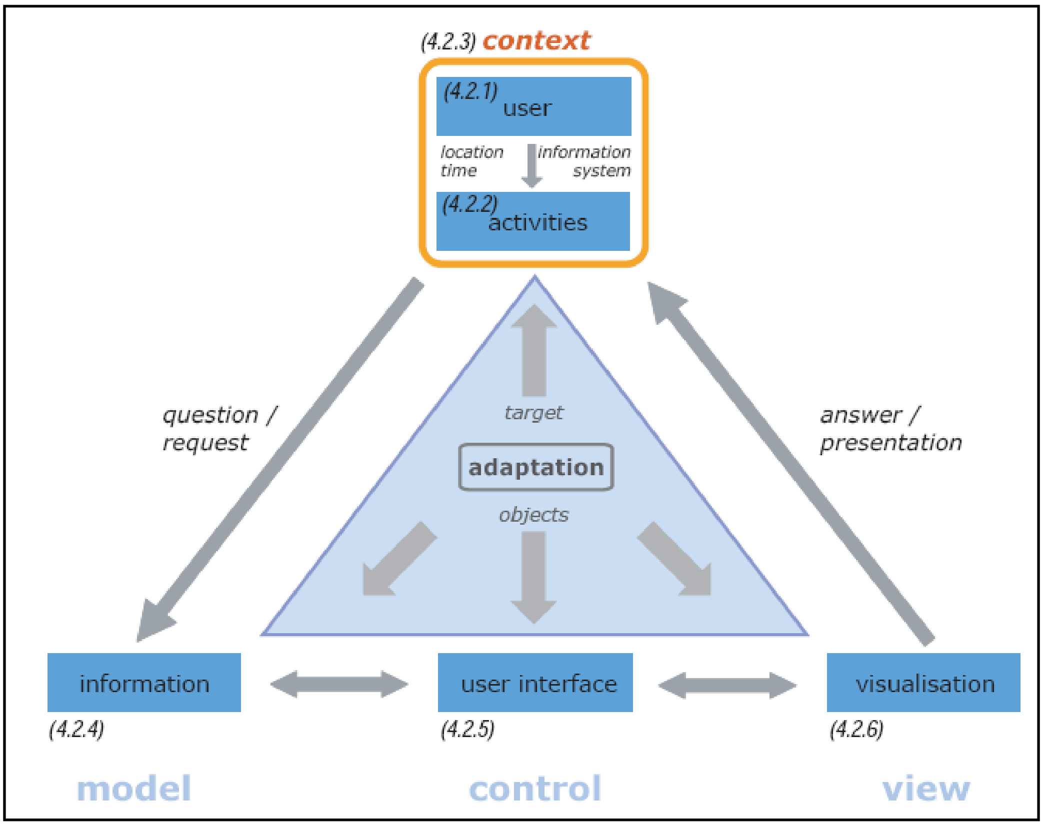

Figure 3.14: Conceptual Framework for Mobile Cartography (Reichenbacher 2004)

Figure 3.15 shows a more detailed view of the adaptation process which lies at the heart of the framework. Adaptations are triggered by changes in the user's context which is measured using five factors, namely situation, user, activity, information and system. If the change in context exceeds some pre-determined threshold, then adaptation may be necessary in one of the three adaptation areas. Once it has been determined that the change in context is significant enough to warrant adaptation, this information is passed to the decision engine. The decision engine decides if adaptation is indeed necessary and decides on an appropriate adaptation strategy. The adaptation engine selects appropriate adaptation methods and applies rules which are obtained from the adaptation model.

The adaptation model contains rules consisting of conditions and actions. The conditions consist of Boolean expressions made up of attributes of the MMV system, while the actions assign new values or expressions to these attributes (De Bra *et al.* 2003). The actual adaptation is executed through the construction of an Adaptor, which applies the chosen methods, parameter values and rules. A broad notion of context is adopted in this framework, including knowledge about users, their current activity and their current situation. Knowledge regarding the user is static, rather than dynamic, and conspicuous by its absence is any form of user modelling (Section 3.8). Despite this shortcoming, this framework covers adaptive MMV in a very detailed fashion.

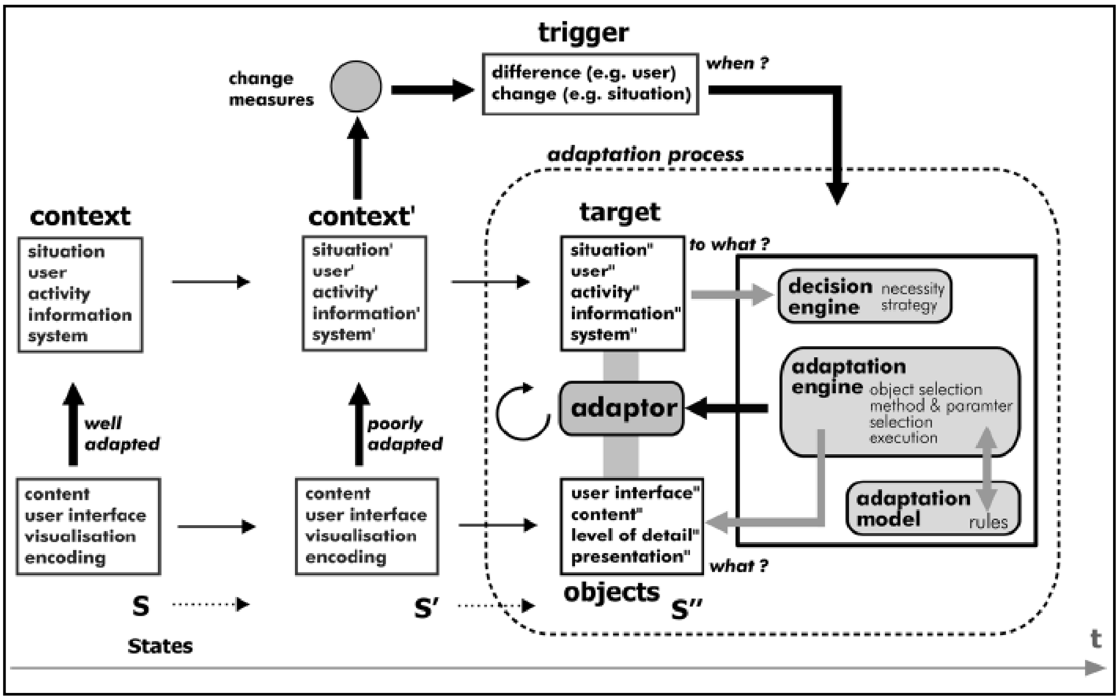

Figure 3.15: Detailed view of the adaptation process which forms a part of the framework in Figure 3.14 (Reichenbacher 2004)

3.11.4 Zipf's Framework

Zipf and Richter (2002) proposed a high-level framework outlining the various steps involved in adapting MMV systems according to users and their context (Figure 3.16). The framework thus meets all three criteria for selection. Zipf's framework regards users' tasks as part of their context. A wide variety of adaptation effects are outlined, most of which fall into the category of visualisation adaptation. This corresponds to the type of AUI outlined in Section 3.9.4.

This framework is merely a high-level conceptual framework discussing the types of adaptation possible and the data necessary to facilitate these types of adaptation. No low level detail is provided regarding the design or working of a system to implement the framework.

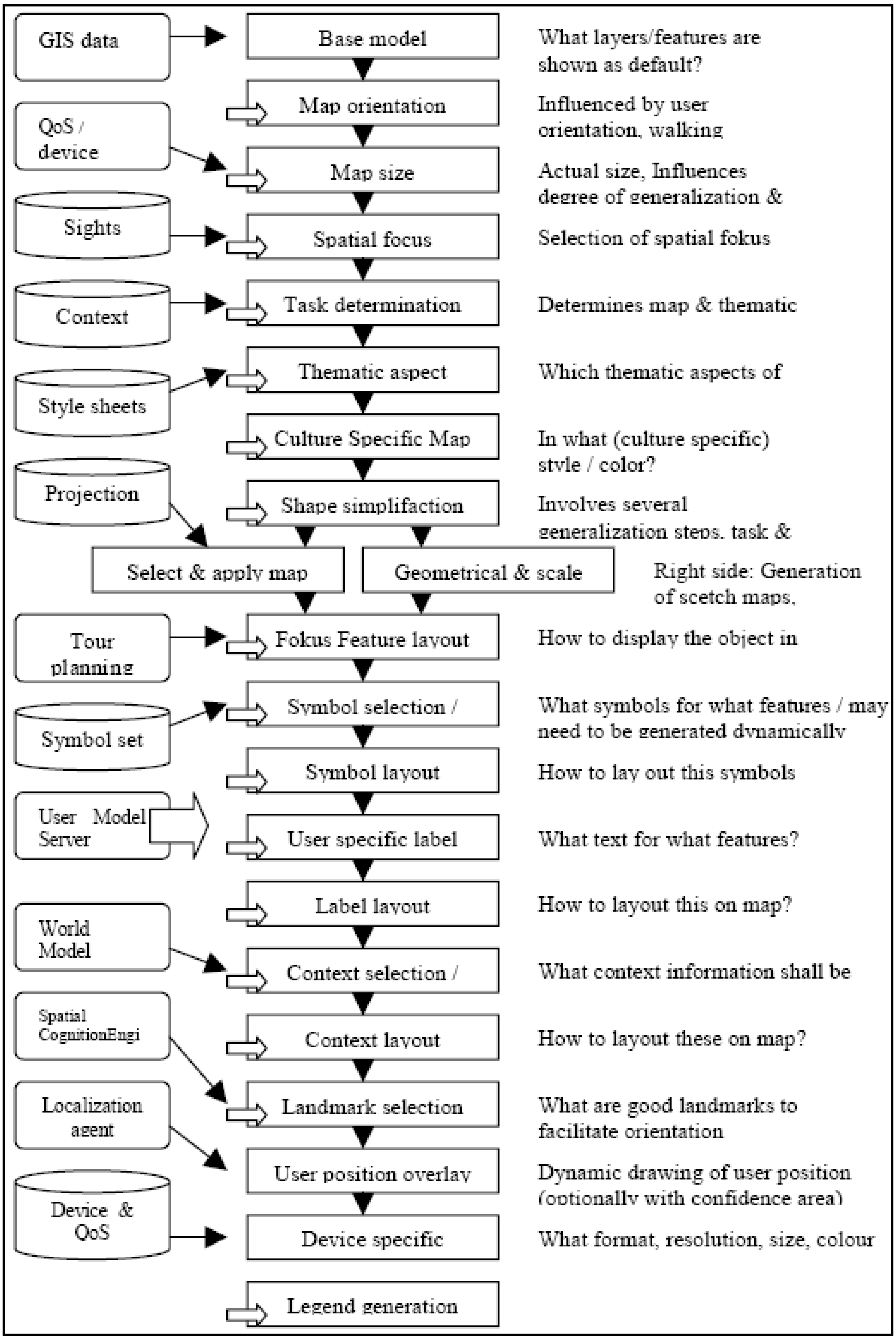

Figure 3.16: Zipf's framework for adapting MMV according to users and their context (Zipf 2002)

3.11.5 Comparison of Models

Three important input variables (user, task and context) relevant to the adaptation of MMV systems were identified in Section 3.10.1. Likewise, three important classes of adaptation effects (information, visualisation and interface) were identified in Section 3.10.2. The existing models discussed in the previous sections are compared in Table 3.2 according to the input variables and adaptation effects supported.

		Adaptive User Interface Models			
		UbiquiTO	Adaptive Geovisualisation Model	Mobile Cartographic Framework	Zipf's Framework
Input Variables	User	✓	✓	✗	✓
	Task	✗	✗	✗	✗
	Context	✓	✗	✓	✓
Adaptation Effects	Information	✓	✗	✓	✗
	Visualisation	✓	✓	✓	✓
	Interface	✗	✓	✓	✗

Table 3.2: Comparison of existing AUI models in terms of input variables and adaptation effects supported

The UbiquiTO architecture and the Mobile Cartographic Framework both support four out of the six input variables and adaptation effects identified in literature. The adaptive geovisualisation model is too abstract to be of practical use in the design of adaptive MMV systems. Furthermore, it only supports three of the six input variables and adaptation effects. Zipf's Framework, while detailed, focuses only on visualisation and map adaptation. The UbiquiTO architecture and the Mobile Cartographic Framework not only consider a wider range of input variables and adaptation effects, but also do so in a more comprehensive manner. The Mobile Cartographic Framework provides a detailed outline of the different types of adaptations which can be performed in an MMV system. The UbiquiTO architecture supports a wide range of input variables and adaptation effects and also outlines specific components which play a role in the adaptation process. The Mobile Cartographic Framework and UbiquiTO architecture were therefore selected as the basis of the model proposed in Chapter 4.

Neither the UbiquiTO Architecture nor the Mobile Cartographic framework cater for all the adaptation input variables and effects identified in Section 3.10.1 and 3.10.2. The UbiquiTO Architecture does not explicitly consider user tasks and does not support adaptation of the user interface. It also targets the limited domain of mobile tourist guide systems. The major limitation of the Mobile Cartographic Framework is that it only focuses on adapting according to changes in the user's context. User modelling is not included in this framework, although it is integrated to a limited degree in the notion of the user's context. Finally, the Mobile Cartographic Framework is a conceptual framework, which seeks to provide a theoretical overview of adaptive mobile cartography. The model proposed in the following chapter seeks to improve on existing models, frameworks and architectures, by supporting all the input variables and adaptation effects of Table 3.2. The proposed model is presented at a lower level than much of the existing work, in order to facilitate the design of adaptive MMV systems.

3.12 Conclusions

AUIs are a subclass of IUIs and rely on user models to improve interaction with the user. They provide a range of potential benefits, from helping ease information overload to assisting users in completing complex tasks. Several problems and shortcomings have also been encountered, mostly stemming from the loss of control and freedom users can experience if AUIs are not implemented correctly. AUIs can be split into three high level groups of components, namely those that observe and record user behaviour, those that turn raw behaviour data into knowledge regarding user preferences and those that perform adaptations based on this knowledge.

AUIs exist in many forms and range from systems that perform adaptations to the user interface to those that provide intelligent help to the user. Many of these forms of adaptation provide potential benefits for addressing the problems of MMV systems. Four types of input variables were identified in existing research into the use of AUIs in MMV. Existing systems adapt to users, their tasks, context and the capabilities of their mobile device. Four types of adaptation effects can also be distinguished. Adaptations can be performed to cater for the differing capabilities of different mobile devices, as well as to the information, visualisation and user interface of MMV systems. Adaptation to cater for different device capabilities (and the corresponding input variable) is beyond the scope of this research.

Existing models, frameworks and architectures incorporate different combinations of these input variables and adaptation effects. These models, frameworks and architectures are, however, either conceptual in nature or only consider a limited set of the input variables and adaptation effects identified. The UbiquiTO Architecture and the Mobile Cartographic

Framework were identified as providing the most comprehensive coverage of the input variables and adaptation effects identified.

A new model is presented in Chapter 4 which is based on the UbiquiTO Architecture and the Mobile Cartographic Framework. This model will address the shortcomings of existing models by catering for adaptations to the information, interface and visualisations in response to the user’s behaviour, tasks and context.

Chapter 4: Model Design

4.1 Introduction

The previous two literature study chapters laid the theoretical foundation for the model to be proposed in this chapter. In Chapter 2, problems with existing techniques for mobile visualisation and MMV were identified. In Chapter 3, the theory of AUIs was discussed and the benefits that AUIs provide for addressing the problems of MMV were identified. Existing AUI models in related domains were identified and reviewed and several shortcomings identified. In this chapter, a new model for adaptive MMV systems is proposed in order to address the limitations of these existing models.

The chapter begins with a motivation for the use of model-based design. The intended audience of the proposed model is identified. The process to be followed in the description of the model is then motivated and outlined. Following this, the proposed model is presented, first in terms of its architecture, followed by a more detailed discussion of the individual components. The interfaces between the various components are then outlined. The supporting data necessary to facilitate the adaptation process is also described. Finally, the algorithm design is discussed, showing how the different model components interact with each other in order to achieve adaptation of the information, visualisation and interface of an MMV system.

4.2 Model-based Design

The word "model" has a variety of different meanings. Two definitions are of particular relevance to the design of software (Collins 1991):

- "*a simplified version of something complex used, for example, to analyse and solve problems or make predictions*"; and
- "*an excellent example that deserves to be imitated*"

These definitions capture two of the essential motivations for developing models. Firstly, models provide a means of describing complex software systems in simpler terms. Secondly, models provide a template for the design of new systems.

Model-based design provides several potential benefits. The model-based design process allows interactive systems to be described in abstract, conceptual terms. This allows designers to focus on key aspects of the design without being distracted by low-level implementation details, particularly early on in the design process (Paternò and Mancini 2000). The use of models in the design of interactive systems provides several benefits, including (Ahmed and Ashraf 2006):

- Management of the increasing complexity of user interfaces;
- Simplified UI architecture leading to better system comprehension; and
- Improved traceability for future maintenance.

In terms of software engineering, models can play a variety of roles. They can be defined in high-level, broad terms to support communication with non-technical stakeholders, at an intermediate level to support system architects and designers, or they can be extremely detailed and low-level to support programmers. In general, models are used to describe the design of a software system and help to abstract away from the technical details (Gomaa and Hussein 2007; Mohagheghi and Aagedal 2007). Sommerville (2006) describes two types of domain-specific architectural models, namely generic models and reference models. Generic models are said to "encapsulate the principal characteristics" of a number of real world systems and may be used in the design of new systems. Reference models are "more abstract" and are used as a means of communicating domain concepts and as a guide in evaluating the quality and completeness of possible designs. The model described in this chapter is similar to the generic model described above, although its design does not only rely on existing systems, but also on relevant models, architectures and frameworks.

Models can be used to describe several different aspects of software systems. Sommerville (2006) identifies four high-level classes of models common to software engineering:

- *Context Models:* Modelling how the system fits into the broader environment;
- *Behavioural Models:* Modelling the behaviour of the system;
- *Structural Models:* Modelling either the architecture of the system or the data processed by the system; and
- *Object Models:* Combine behavioural and structural modelling.

The model described in this chapter relies largely on object modelling techniques, based on UML, to describe behavioural and structural aspects of an MMV system incorporating an AUI.

4.3 Intended Audience

The intended audience for the model to be presented in this chapter consists of the designers and developers of adaptive MMV systems. The integration of AUIs into MMV systems is becoming more common as designers realise the benefits that can be achieved (Section 3.10). Existing solutions, however, tend to be implemented in an ad-hoc manner and as a result, any benefits achieved through adaptation are difficult to replicate. While it is impossible to provide an exact design template for the creation of every type of adaptive MMV system, the model presented in this chapter will seek to describe the design of such systems at a level of detail low enough to identify the major system components necessary to implement an adaptive MMV system, as well as the interfaces and communication between these components. The implementation of these individual components can then be tailored according to the application type and domain. Such a model is potentially also of great value to researchers, seeking to compare different AUI techniques in MMV systems. Different implementations of individual components can be evaluated in order to compare different techniques for the various aspects of adaptation (e.g. different user modelling techniques). A proof-of-concept prototype, based on the proposed model, is presented in Chapter 5 to illustrate how the model can be used to successfully implement an adaptive MMV system. The rest of this chapter is devoted to detailing a new model for adaptive MMV. In the next section, the process to be followed in the description of the model is presented.

4.4 Model Design Process

Olivier (2004) suggests that the process of designing a model can be conducted similarly to the process of designing a software system. The approach to model design followed in this chapter is a modified version of the software design process described by Sommerville (2006). Sommerville identifies the following activities as being part of the software design process:

- *Architectural design*: The sub-systems which comprise the system are defined and the relationships between the sub-systems identified.

- *Abstract specification*: The services provided by each sub-system are described in more detail and any constraints under which each sub-system must operate are identified.
- *Interface design*: For each sub-system, the interfaces between it and other sub-systems are identified. This allows each sub-system to communicate with other sub-systems without any knowledge of these sub-systems' inner workings.
- *Component design*: The individual components (a sub-system can comprise several components) are allocated services and the interfaces to other components are defined.
- *Data structure design*: The data structures used in the system implementation are described.
- *Algorithm design*: The algorithms to be used by the system are designed and specified.

Some of these aspects are more suited to the design of a specific system than a model. For example, the data structure design (specifically of the data to be visualised) is something that is (to a large degree) specific to the application domain. As a result, the data design will be restricted to data that is necessary to support the adaptation process. Furthermore, the specification of different parts of the model will take place at component level only, rather than at sub-system level, as intra-system communication will take place at component level.

The discussion of the proposed model (called the Proteus[7] Model) largely mirrors the above outline, starting with a high-level overview of the architecture of the model. This is followed by a lower-level discussion of the individual components and the interfaces between these components. Finally, the data and algorithm design of the model are described.

4.5 Model Architecture

The architecture of the Proteus Model is shown in the form of a block diagram in Figure 4.1. The model draws a large measure of its design from the UbiquiTO Architecture (Section 3.11.1) and the Mobile Cartographic Framework (Section 3.11.3). The Proteus Model is proposed to address the shortcomings of existing models by facilitating adaptation of the

[7] In Greek mythology, Proteus, son of Poseidon was able to change his shape at will

information, visualisations and user interface in response to the user's behaviour, tasks and context. The analysis of existing models conducted in Section 3.11.5 revealed that no model currently exists which combines all these input variables and adaptation effects into a single model. The three types of adaptation facilitated by the Proteus Model are drawn from the Mobile Cartographic Framework. These three types of adaptation are:

- *Information Adaptation:* adapting the information selection, organisation and structure;
- *Visualisation Adaptation:* adapting the presentation of information, including aspects such as level of detail and location; and
- *User Interface Adaptation:* adapting the user interface in terms of mode, style and the availability and ordering of options.

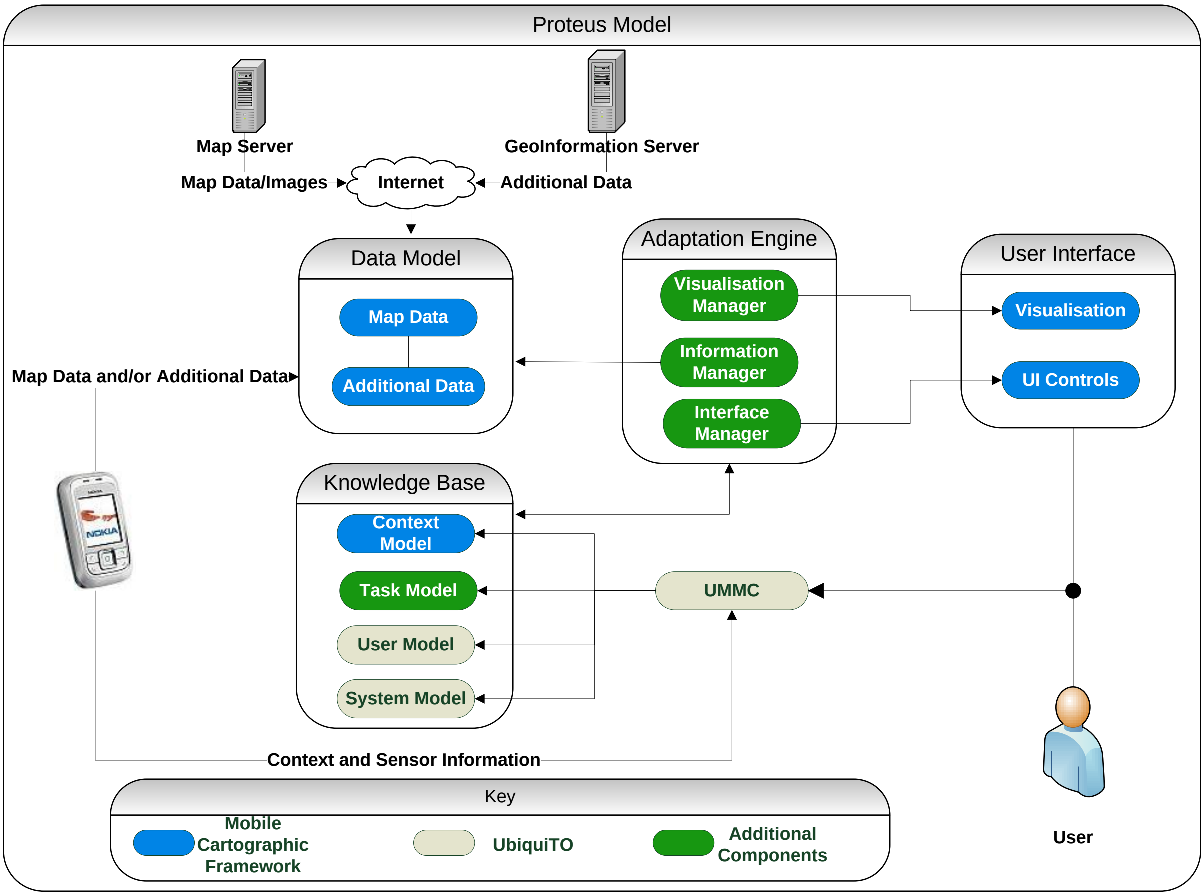

Figure 4.1: Proteus Model Architecture

Adaptation in the Proteus Model is supported by three components, namely the User Monitoring and Modelling Component (UMMC), the Knowledge Base and the Adaptation Engine. The UMMC is responsible for recording and modelling user interaction, in order to infer users' preferences from their past behaviour. The Knowledge Base contains several models which contain knowledge regarding the user (including preferences and context) and the system (including system tasks and the status of adaptable parameters). The Adaptation Engine contains components to manage the adaptation process, consulting the models in the Knowledge Base before making changes to the Data Model, Visualisation and User Interface (UI) Controls. The UMMC (in recording user interaction) and the Knowledge Base implement the afferential AUI component (Section 3.7), the UMMC (in modelling user interaction) implements the inferential AUI component (Section 3.8) and the Adaptation Engine implements the efferential AUI component (Section 3.9). The individual components of the Proteus Model are discussed in more detail in the following section.

4.6 Component-Level Design

Components in the Proteus Model are drawn from the Mobile Cartographic Framework, the UbiquiTO Architecture, or are additional components intended to address the limitations of existing models. Some components are standard parts of MMV systems and will not be discussed in detail. These include the User Interface, which is split into the Visualisation and UI controls which allow the user to manipulate the Visualisation. The Data Model is split into Map Data and Additional Data. Map Data refers to the maps themselves. Maps in MMV systems can be in either vector or raster (image) format (Burigat and Chittaro 2005). Vector format requires low-level map data describing map features to be stored and used to render the maps. This is more computationally expensive than using raster maps, but provides the advantage of increased flexibility in terms of how the maps are rendered. Raster maps are often downloaded via the Internet, with the MMV system merely supplying the required latitude and longitude in order to retrieve the appropriate raster map tiles.

MMV systems typically display more than just plain maps. The Additional Data in the Data Model refers to the data that MMV systems visualise in the context of the underlying maps. This data is typically spatial or spatio-temporal in nature, although map metaphors are sometimes used to visualise non-spatial data (Section 2.3). The remaining components interact with these standard MMV components and drive the adaptation process. These components will be described in the sections that follow.

4.6.1 User Monitoring and Modelling Component (UMMC)

The UMMC combines the "Watcher" component of the UbiquiTO architecture with a component for performing user modelling. This is one of the most crucial components in the adaptation process, observing all user interaction with the MMV system. This information is used to learn user preferences, identify the user's task (with the help of the Task Model) and perform updates to the System Model where necessary. Before updates are made to the User Model, data is put through a user modelling process in order to turn raw user interaction data into knowledge regarding user preferences. The UMMC is therefore responsible for implicitly acquiring user preferences (Section 3.7).

4.6.2 System Model

The System Model borrows elements from both the Mobile Cartographic Framework and UbiquiTO architecture, but follows a more traditional AUI approach (Section 3.6). This model stores the adaptable parameters of the system and the current status of these parameters. The System Model is an essential AUI component, particularly as both the user and the system are able to make modifications to the adaptable parameters. A central repository is therefore needed in order to store the current status of these parameters and avoid inconsistencies. The System Model is therefore a dynamic model, constantly updated by the UMMC to reflect the current system status and changes performed by either the system or the user.

4.6.3 Task Model

Task models are an essential AUI component (Section 3.6). Despite this, neither the UbiquiTO architecture nor the Mobile Cartographic Framework explicitly defines a task model. In the Mobile Cartographic Framework, users' current activities are considered as part of their context and modelled as such (dynamic task modelling, Section 3.7.1). The Task Model in the Proteus Model follows a more traditional AUI approach, modelling the different user tasks and the steps involved (static task modelling, Section 3.7.1). As a result of the limited interaction techniques available on mobile devices, tasks in MMV systems are generally fairly linear. Simple notations should therefore be sufficient to encode these tasks in the Task Model. The Task Model is consulted in order to determine the user's current task. User interaction data (which is monitored by the UMMC) can be compared to known system tasks stored in the Task Model in order to identify the user's current task. The user's task influences the type of adaptations that need to be performed.

4.6.4 User Model

User modelling is found to some degree in both the UbiquiTO architecture and the Mobile Cartographic Framework. In the Mobile Cartographic Framework, however, a user model is not explicitly defined. As a result, the User Model in the Proteus Model is drawn from the UbiquiTO architecture (Figure 4.1). Along with the Task Model, the User Model is an essential component of any AUI (Section 3.6). The output from the UMMC is stored persistently in the User Model (dynamic user modelling, Section 3.7.4). The User Model is consulted when any adaptation is performed in order to determine the relevant user preferences. This ensures that the interface, information and visualisation adaptations are aligned with the user's needs and behaviour.

4.6.5 Context Model

Context is an important driver of adaptation in MMV systems (Section 3.10.1). Context information is primarily obtained from sensors on the mobile device, which can measure and record information such as the time, the user's location and user movement. In the Proteus Model, this information is stored in the Context Model, which is dynamically updated as necessary. The user's current task, as determined from the Task Model, is also stored in the Context Model. This information is used to inform the adaptation process, enabling context-sensitive adaptations to be performed. The Context Model is also responsible for initiating adaptation. When a new user task is detected (and the Context Model updated to reflect this), the manager components which perform the relevant adaptations are invoked by the Context Model (where necessary).

4.6.6 Visualisation Manager

The Visualisation Manager is the first of three manager components in the Proteus Model and is responsible for all adaptations to the visual presentation of information. MMV systems present information to the user in a map-based format and include several parameters that can be adapted (Section 3.10.2). The Visualisation Manager acts as a central point of control for making adaptations to the presentation of information. All adaptations are performed in consultation with the relevant models in the Knowledge Base. The User Model is of primary importance, as it is important that the visualisation is rendered according to the individual user's preferences. The Context Model is also potentially important as several visualisation adaptations may be performed in response to changes in the user's context. The Task Model allows the system to identify when a new visualisation is to be rendered, allowing this visualisation to be adapted according to user preferences.

4.6.7 Information Manager

The Information Manager is responsible for managing adaptation to the selection and organisation of the information being visualised by the MMV system. This could be either the Map Data itself (e.g. the selection of which map features to display) or the Additional Data which is visualised in a map-based context. As with the Visualisation Manager, all adaptations are performed in consultation with the models in the Knowledge Base. The Context, Task and User Models play an important part in the adaptation process. For example, the currently visualised information could be filtered according to the user's location.

4.6.8 Interface Manager

The Interface Manager is responsible for making changes to the interface presented to the user (excluding the visualisation itself). This could be in the form of shortcuts provided to the user, the availability of functions or the ordering of menu options. The Knowledge Base is consulted in order to ensure that adaptations performed are in accordance with the user's task, context and preferences. For example, if the user is performing a navigation task, providing a shortcut to recently selected destinations could improve the efficiency with which users are able to complete this task.

4.7 Interface Design

The next step after identifying the major components of the Proteus Model is to define how these components interact with one another. To do this, the interfaces between the different components need to be specified. Figure 4.2 shows the interfaces between the different components in the Proteus Model at a lower level than the architecture diagram in Figure 4.1. Note that the System Model and Context Model methods provided are by no means exhaustive, as these will vary depending on the application domain, device capabilities and the tasks that the system facilitates. These are included merely to illustrate the possible methods that could be included. Significant relationships and relatively high-level interface methods are depicted. For the sake of readability, container components are shown as taking part in relationships where all sub-components are involved in relationships of the same nature with another component. For example, the Context Model can invoke all three manager components, but is shown in Figure 4.2 to interact with the parent component, the Adaptation Engine.

Several high-level relationships are depicted in Figure 4.2. The UMMC is responsible for performing updates to the different models in the Knowledge Base. The components of the Adaptation Engine can be invoked directly from the UMMC, or as a result of a detected change in the user's task (as recorded in the Context Model). The components of the Adaptation Engine query the different models of the Knowledge Base in order to ensure that the appropriate adaptations are performed. Information, visualisation and interface adaptation can then be performed, resulting in changes to the Data Model, Visualisation and UI controls. These changes are recorded in the System Model to ensure a consistent system state is maintained.

4.8 Data Design

Any system based on the Proteus Model will incorporate two different sets of data. Firstly, the system will include data which is stored in the system's Data Model and which is visualised for the benefit of the user. Secondly, the system will contain data which is stored in the various models of the Knowledge Base and which is used to inform the adaptation process. The data design of the individual models will now be outlined. Extensible Markup Language (XML) will be used to show the schema of the various models. XML is a human and machine readable format, which allows model schemas to be described in a simple, understandable format. Furthermore, XML parsing is supported by major mobile development frameworks, such as the .Net Compact Framework and Java Micro Edition, so the schema shown below can be directly translated into actual implementations to be used by MMV systems. It is important to note that MMV systems differ widely and different systems will require different parameters to be adapted. The schemas shown below will describe the models in as comprehensive a manner as possible, including the data necessary to facilitate a wide range of adaptations in the areas within the scope of the model. Any system implementing the System, Task, User and Context Models will therefore only implement a subset of all the attributes shown. Some systems may incorporate additional dimensions. The schema will attempt to cover those attributes which have been identified in existing literature as important in the adaptation process (Section 3.10.2).

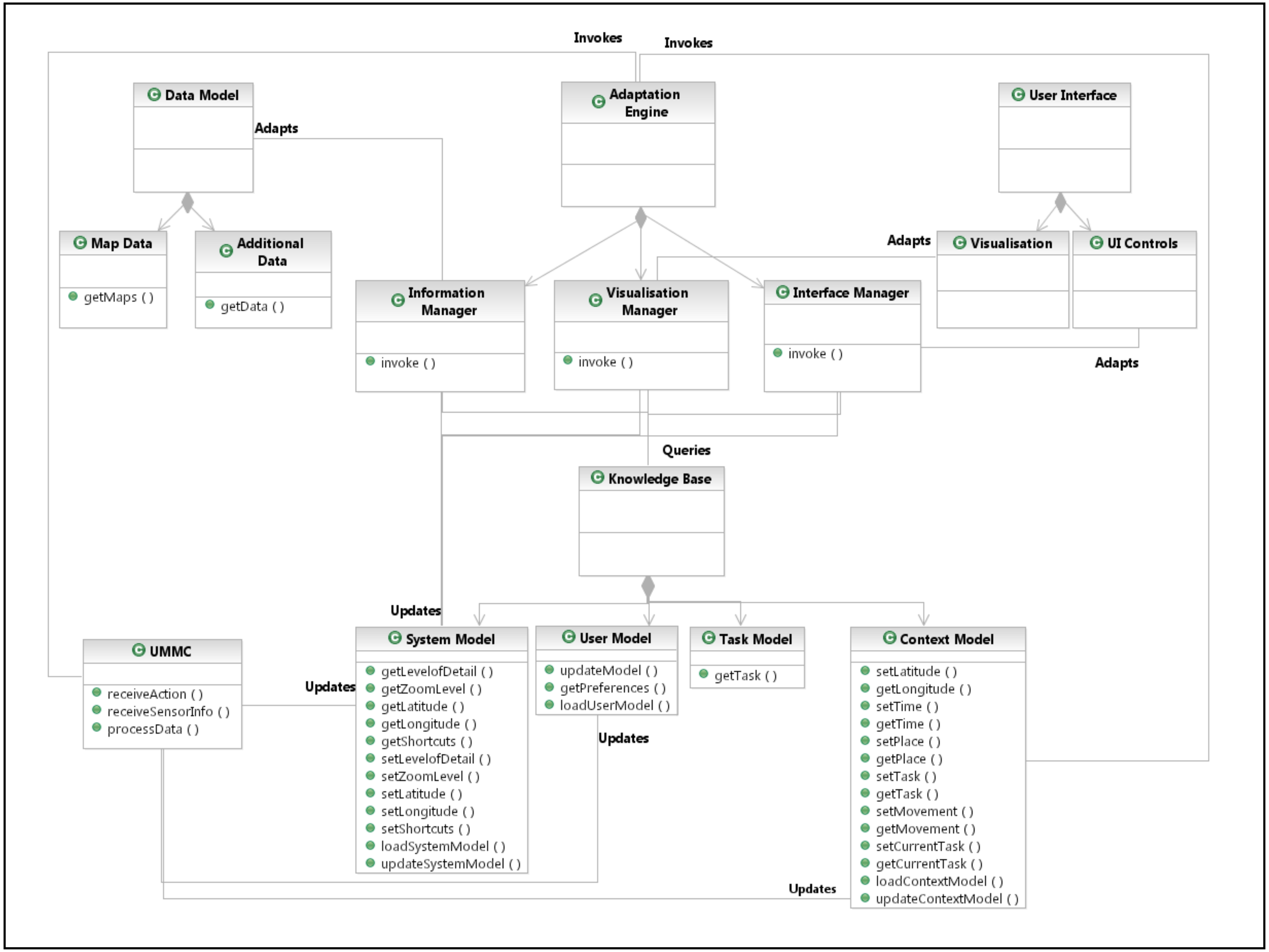

Figure 4.2: Interface between components

4.8.1 System Model

The System Model acts as a central repository for storing the current values of the adaptable parameters of the system. Given the wide range of adaptation effects possible (Section 3.10.2), a wide range of values exist that may need to be stored in the System Model. The schema shown in Figure 4.3 divides the data stored into information, interface and visualisation parameters and is drawn to a large degree from Reichenbacher (2004).

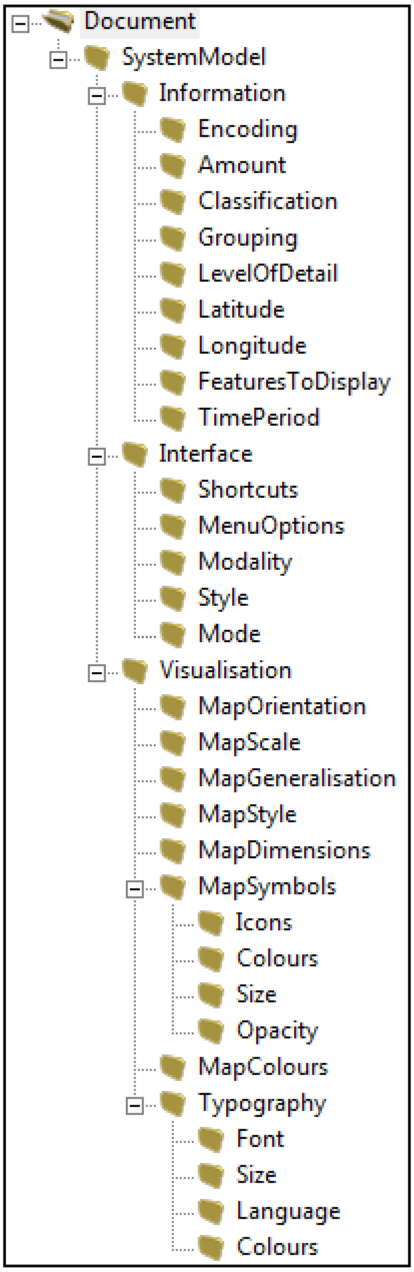

Figure 4.3: System Model Schema

4.8.2 Task Model

The Task Model is the only static model in the Proteus Model and is used to model tasks that the user can perform with the MMV system. Its schema, as shown in Figure 4.4, is very simple. Tasks are split into sub-tasks hierarchically. The limited interaction possibilities provided by mobile devices generally result in tasks that can be decomposed into linear sets of subtasks. Each task also has a unique identifier, which could be either a code or a more meaningful name. As a static model, it is possible to hard-code the Task Model into an MMV system for the sake of efficiency. Retaining an external representation in XML form does, however, hold benefits in terms of the extensibility of the system. The Task Model should be indexed to provide for efficient identification of user tasks based on low-level interaction data (given a user action, a list of possible system tasks should be returned). A short history of user interactions may need to be stored to differentiate between tasks featuring common sub-tasks.

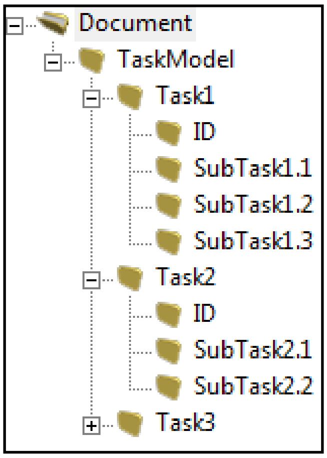

Figure 4.4: Task Model XML Schema

4.8.3 User Model

The User Model is the most complex model in the Proteus Model. It is a highly dynamic model and is updated by the UMMC, storing the user's preferences which are determined by the UMMC through user modelling. The structure of the User Model is closely linked to that of the System Model. Whereas the System Model stores the current state of the different parameters, the User Model stores the user's preferences for these parameters. In the Proteus Model, these preferences are stored for different user tasks (Figure 4.5), as users' preferences for one task may differ significantly from their preferences for another. An alternative User

Model schema could be used to store user preferences for different data sets, contexts or visualisation formats, or combinations thereof.

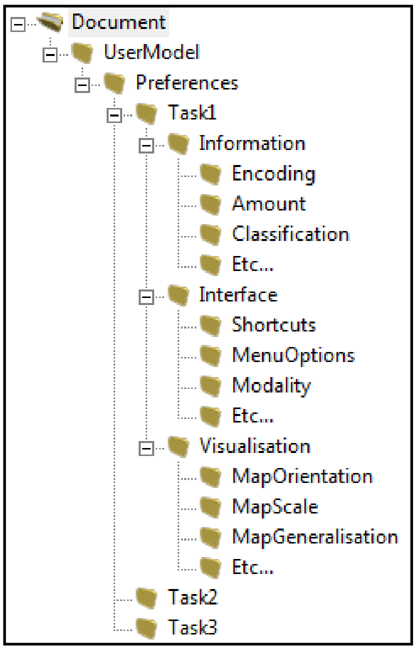

Figure 4.5: User Model Schema

4.8.4 Context Model

The Context Model contains information obtained from various sensors, including information regarding the time, the user's location and the user's current activity (Section 2.4.1). The context information that can be obtained also depends on the device's capabilities. As more advanced sensors are integrated into mobile devices, it is possible that more context information may be obtained. Also stored in the Context Model is the user's current task. This allows the system to adapt according to what the user is currently doing. A hypothetical Context Model schema is shown in Figure 4.6.

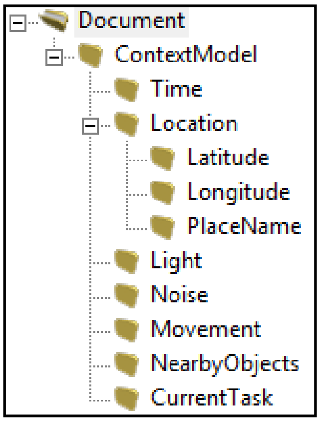

Figure 4.6: Context Model Schema

4.9 Algorithm Design

The following major AUI components were identified as being part of the adaptation process (Section 3.6):

- *Afferential Component:* Collects data about users and their context;
- *Inferential Component:* Converts this data into useful knowledge about users; and
- *Efferential Component:* Uses this knowledge to perform adaptation.

The algorithm design of each of these three components is discussed in the sections that follow. The interaction between the different components of the Proteus Model which combine to implement each of these three components is discussed.

4.9.1 Afferential Component

The UMMC observes user interaction with the system. MMV systems typically provide one or two (in the case of stylus/keypad devices) input modalities. User interaction data is generally processed at a fairly low-level in the source code. Events are usually only available to the programmer with low-level descriptions, such as a key press (with a unique key code associated), or a screen selection at certain co-ordinates (on touch screen or stylus-based devices). It is up to the programmer to then translate this low-level interaction data into more meaningful information about the action the user has just performed. For example, the fact that the user has pressed the left soft key is meaningless if considered in isolation. If, however, it is known that the left soft key corresponds to the "Zoom in" function, this low-level interaction information becomes more meaningful. The System Model, which stores the current values of different system parameters, can assist with interpreting low-level interaction data. Therefore, some pre-processing needs to be embedded into the system in order to turn low-level interaction data into meaningful user actions. The Task Model can be used to identify the high-level task the user is currently performing. This is important because (as shown in the User Model schema in Figure 4.5) user preferences may differ from one task to another. The user's current task is stored in the Context Model (several high-level tasks for MMV systems were identified in Section 2.5.2).

User interaction data is not the only data that is recorded in the Proteus Model. Sensor information must also be obtained and stored in order to update the Context Model. Context information may also require pre-processing. For example, the system may use place names rather than latitude and longitude information, so co-ordinates may first need to be converted into the nearest known place name.

User interaction data can, in theory, be input straight into the UMMC in order to update the User Model. This approach can, however, negatively affect system performance as the system must perform user modelling while still providing a responsive user interface. Given the constraints of mobile hardware (Section 2.4.2), this is likely to be problematic. It is also unnecessary, as it is unlikely that a single action would result in a major change to the preferences stored in the User Model. As a result, it is recommended that the interaction data be stored in a persistent or semi-persistent data store (such as a text file or XML file) and that updates to the User Model (through the UMMC) be performed at system shutdown. This will ensure that system performance is not negatively affected by the UMMC monopolising CPU time. Figure 4.7 illustrates how the UMMC would typically work.

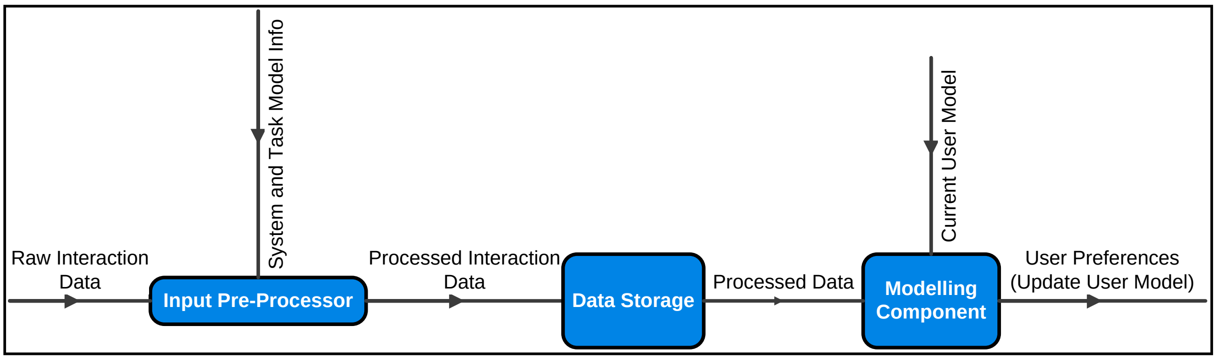

Figure 4.7: The functioning of the UMMC

4.9.2 Inferential Component

User modelling is necessary in order to convert user interaction data into useful knowledge regarding the user's preferences. User modelling and the available techniques were discussed in Section 3.8. No generally accepted user modelling technique exists for MMV. Several variables need to be considered, including the inputs into the user modelling process and the types of adaptations the system needs to facilitate. Another issue which needs to be considered is whether user modelling will be performed on the device itself or on a remote server, with the results downloaded to the mobile device (thin client architecture). It is therefore difficult to prescribe a user modelling technique that will suit every situation. In the Proteus Model, the UMMC will be considered a black-box component, converting input data into user preference data, such as that outlined in the System Model and User Model schemas (Figure 4.3 and Figure 4.5). The "Modelling Component" in Figure 4.7 represents this black box component. A user modelling technique can be implemented which suits the requirements of the particular MMV system being developed.

4.9.3 Efferential Component

The timing of adaptation is an important issue that can negatively affect the usability of an AUI. In Section 3.9 three possibilities were identified, namely before, during or between usage sessions. It is generally accepted that making frequent, arbitrary changes can make the system very confusing for the user (Kules 2000), (Dieterich *et al.* 1993). Given the limited computer experience of many users of mobile devices, this problem is likely to be more pronounced in mobile systems. Hence, it is advisable to keep adaptations to a minimum. As a result, it is recommended that adaptations in the Proteus Model only occur when the system starts up, or when the system is performing a new task (when the interface is likely to change significantly anyway). In this way, adaptations will result in minimal disruption to the user. Adaptation will typically be performed when a new visualisation is to be rendered, minimising confusion and ensuring that the resulting visualisation corresponds to the user's preferences.

Figure 4.8 shows a recommended approach to the timing of the different processes involved in adapting an MMV system. The tasks from Figure 4.7 that are resource intensive are performed at system start-up (updates to the User Model) and system shutdown (recording user interaction data and Knowledge Base models). This ensures minimal disruption to normal functionality. While the system is operating, user interaction is continuously observed and logged, in order to identify when the user is performing a new task. The manager components are then invoked as necessary in response to user behaviour.

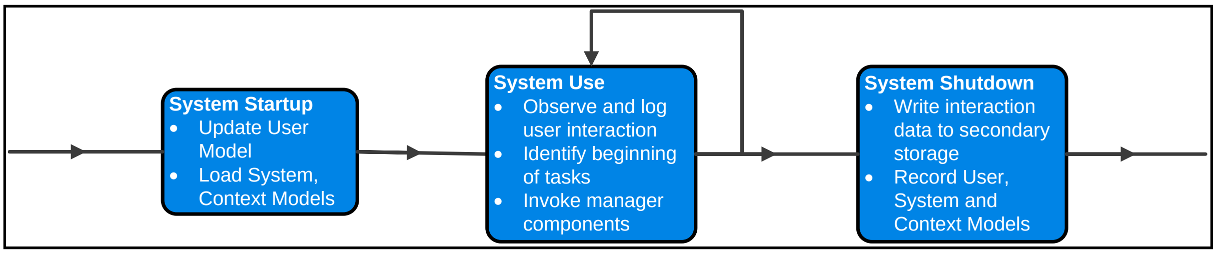

Figure 4.8: Recommended adaptation timing for systems based on the Proteus Model

The Proteus Model supports three different types of adaptation, namely information adaptation, visualisation adaptation and interface adaptation. Each of these forms of adaptation will now be examined in more detail, including the interaction between different components that needs to take place in order to achieve the adaptations.

Figure 4.9 shows the sequence of component interactions involved in information adaptation. The sequence diagram shows how user interaction is monitored by the UMMC. The System Model is updated to reflect changes to the adaptable system elements. The Task Model is consulted in order to identify whether the current task has changed. If it has, the current task is updated in the Context Model which invokes the Information Manager. If adaptation is required, the Information Manager queries the User Model to determine the user's information preferences for the current task. Information Adaptation is then performed and can involve anything from filtering the information selection, to adapting how information is structured. The System Model is updated to reflect any adaptations which are performed.

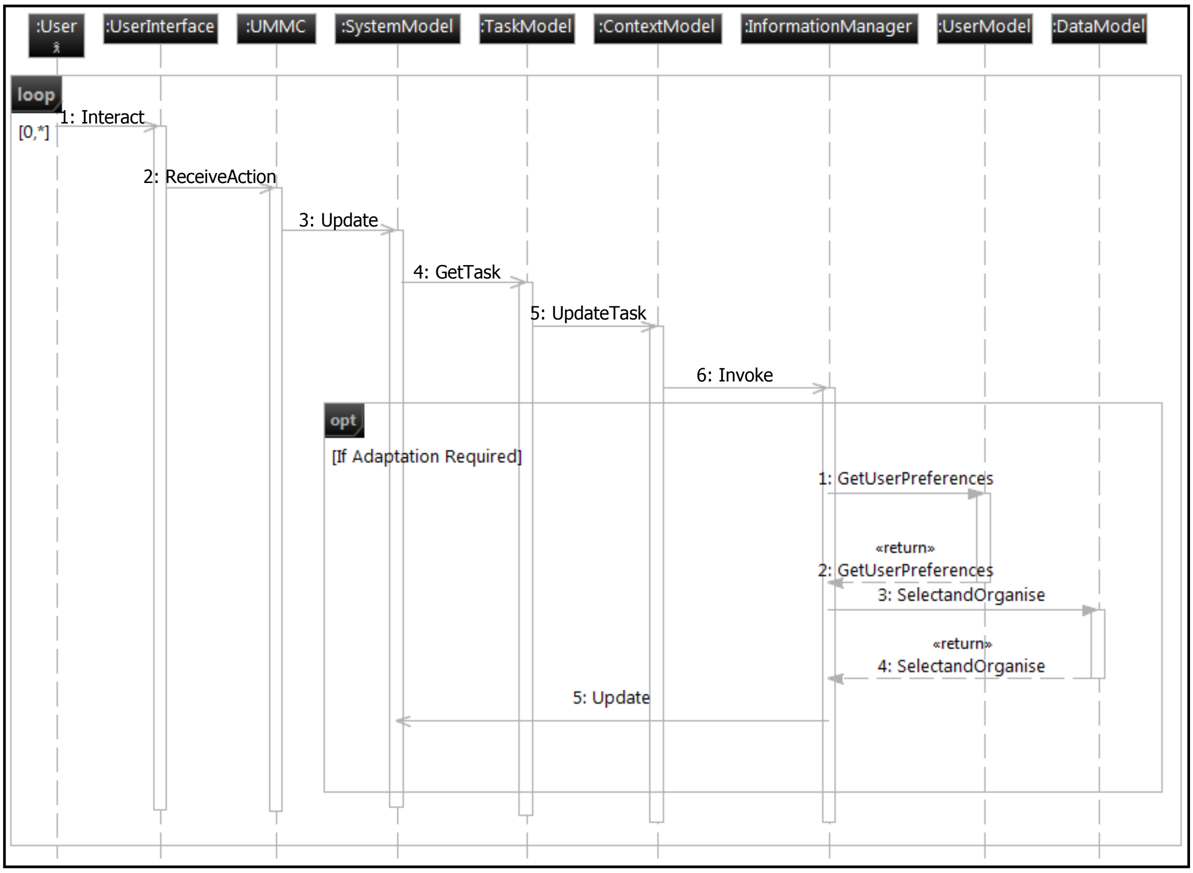

Figure 4.9: Sequence diagram showing how information adaptation is performed

Figure 4.10 depicts the functioning of the Visualisation Manager. The Visualisation Manager works similarly to the Information Manager. User interaction is continually monitored to determine when/if a new task has begun. Once this has been determined, the Context Model is updated and the Visualisation Manager is invoked. If adaptation is necessary, the

Visualisation Manager consults the User Model in order to ensure that the Visualisation rendered conforms to the user's preferences given the current user task. The Visualisation Manager then sends any changes that were made to the System Model which is updated to ensure that a consistent system state is maintained.

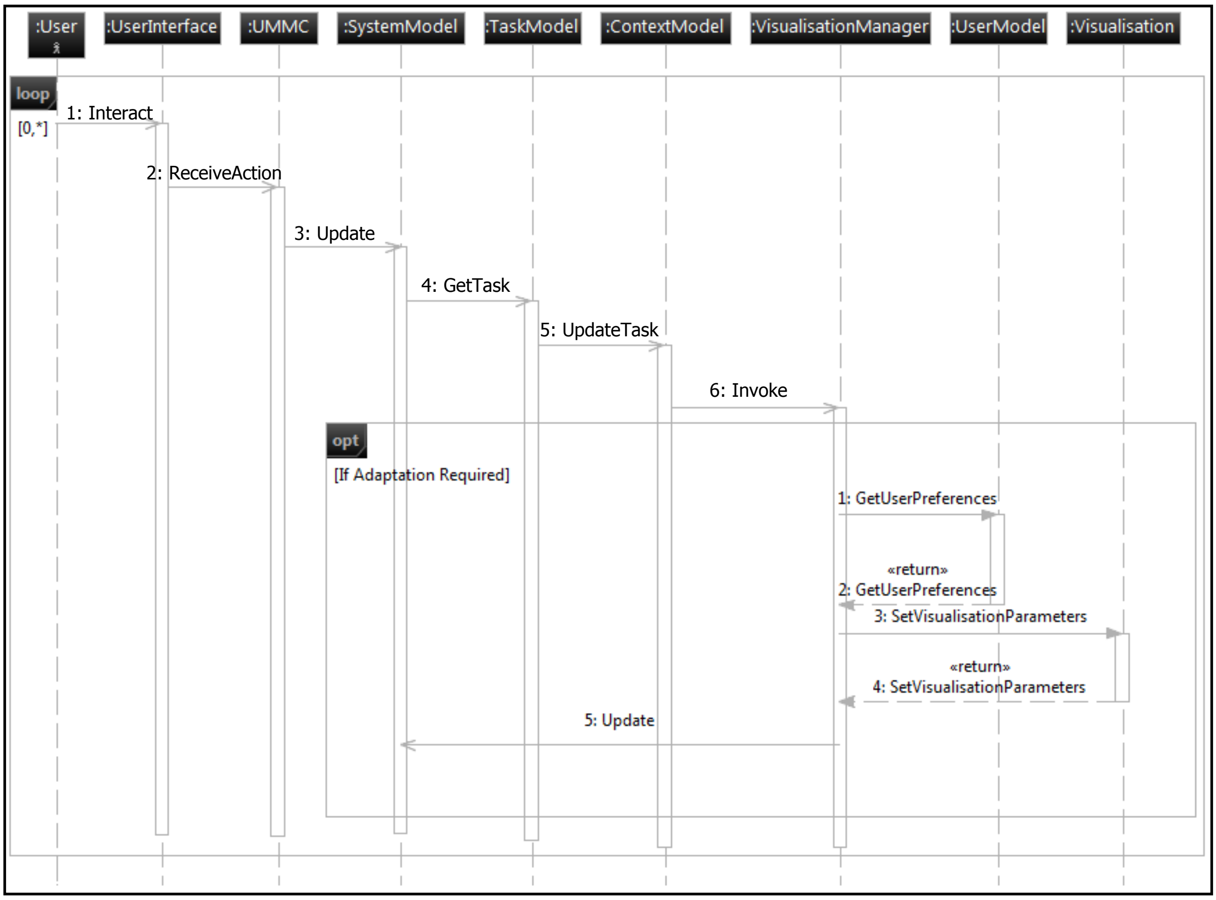

Figure 4.10: Sequence diagram showing how visualisation adaptation is performed

The component interactions involved in the functioning of the Interface Manager are shown in Figure 4.11. The Interface Manager works similarly to the other two manager components in most respects. One significant difference is, however, evident. Interface adaptations are not restricted to the start of new tasks, as the user may access a screen with menu options or shortcut icons at any moment. As a result, interface adaptations such as reordering shortcuts and menu options, or making options available or unavailable, may be required at any moment. Accordingly, the invocation of the Interface Manager is not limited to the beginning of a new task. It can also be invoked by the UMMC when the user's interaction data reveals that a system state has been entered which requires interface adaptation. Alternatively, the Interface Manager may be invoked from the Context Model in a similar fashion to the Information and Visualisation Managers.

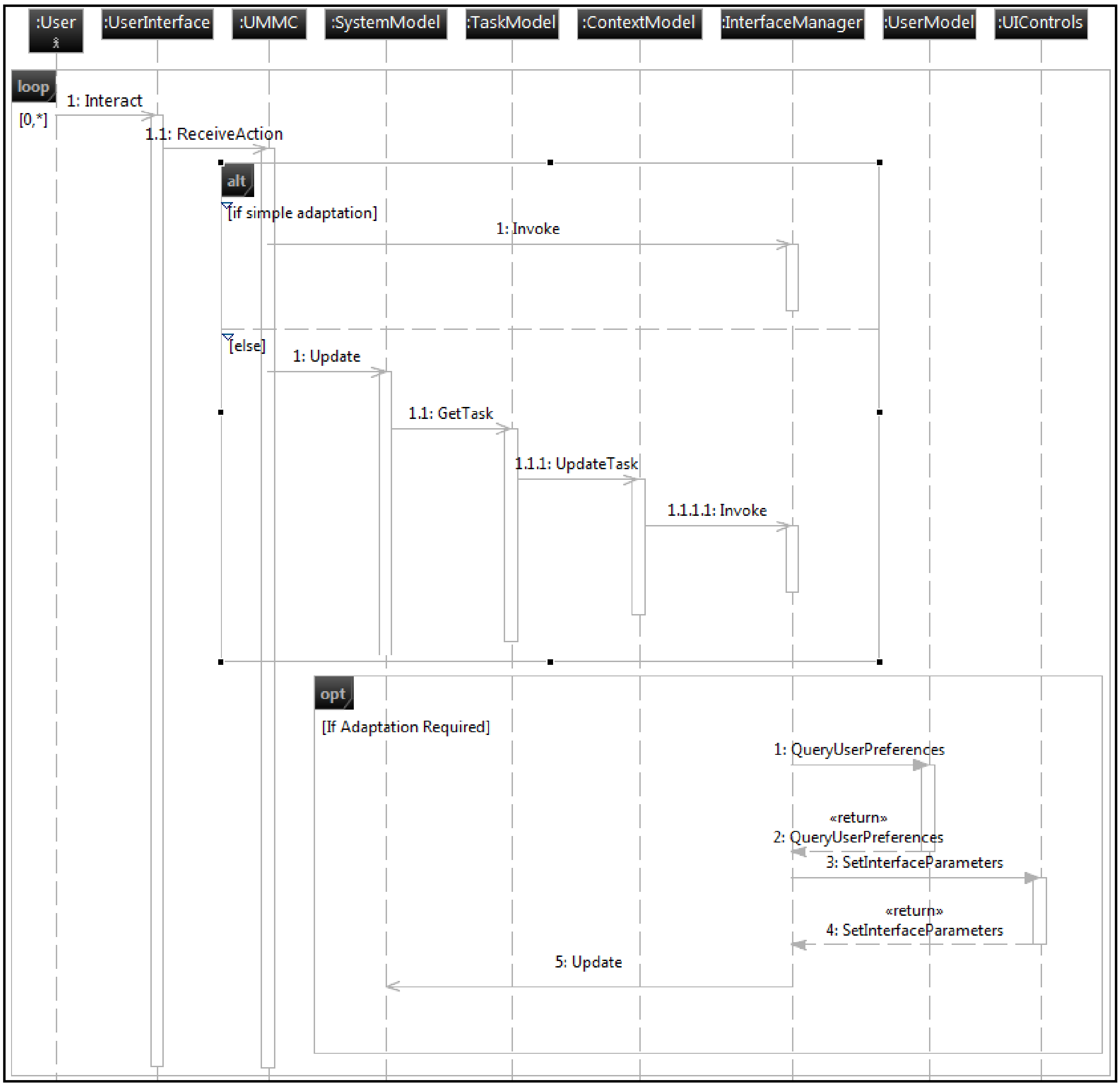

Figure 4.11: Sequence diagram showing how interface adaptation is performed

4.10 Conclusions

The Proteus Model presented in this chapter provides a template for the design of adaptive MMV systems. The Proteus Model incorporates several components from the UbiquiTO Architecture and the Mobile Cartographic Framework. The shortcomings of these models are addressed by incorporating additional components to facilitate information, visualisation and interface adaptation in response to user behaviour, tasks and context. The Proteus Model describes the architecture of an adaptive MMV system, with three high-level components managing the adaptation process. The Knowledge Base contains four different models, enabling the design of MMV systems which are able to adapt the system to user behaviour, tasks and context. The UMMC is responsible for observing and modelling user behaviour and for performing updates to the models in the Knowledge Base. The Adaptation Engine is responsible for performing adaptation, facilitating a wide range of adaptations through the use of three components which manage information, visualisation and interface adaptation. The interface design of the Proteus Model demonstrates how these components interact with one another in order to perform adaptation.

The data design of the Proteus Model was presented, motivating the use of XML to store the contents of the different models in the Knowledge Base. XML is a simple format which is easy to implement and is supported by major mobile development platforms. Example schema were presented for each of the four models, illustrating the typical attributes that would need to be stored to enable the different forms of adaptation supported by the Proteus Model.

The algorithm design of the Proteus Model demonstrates how the UMMC, Knowledge Base and Adaptation Engine work together to adapt MMV systems in terms of the information visualised, the map-based visualisations presented to the user and the supporting user interface.

The Proteus Model addresses the shortcomings of existing models by incorporating the major input variables and adaptation effects identified in Section 3.10. Unlike many of the existing models, the Proteus Model is not merely conceptual in nature, but describes the essential aspects of the design of an MMV system.

The implementation of a prototype MMV system based on the Proteus Model is described in Chapter 5.

Chapter 5: Implementation

5.1 Introduction

The design and implementation of a prototype adaptive MMV system, based on the Proteus Model is described in this chapter. This prototype, called MediaMaps, was developed as a proof-of-concept of the model which was outlined in the previous chapter. The implementation of the prototype will provide a concrete example of how the different components of the Proteus Model can be implemented and how these components interact with each other in order to achieve adaptation. MediaMaps was also developed as a tool to evaluate the application of AUI techniques in an MMV system.

This chapter begins with a motivation for the application domain selected for the proof-of-concept prototype. The software tools and development methodology used for implementation are motivated and described. The implementation of MediaMaps is then described, illustrating how the architecture and individual components of the Proteus Model were implemented. The different forms of adaptation that were implemented in MediaMaps are discussed in detail.

The design of the Proteus Model and the implementation of MediaMaps as a proof-of-concept of this model were published in the proceedings of SAICSIT 2008 (van Tonder and Wesson 2008).

5.2 Selection of Application Domain

Several criteria were considered in selecting a domain for the proof-of-concept prototype. An application domain was required which corresponded to the domain of the Proteus Model. The following criteria were therefore used:

- The prototype had to incorporate MMV as a central part of its functionality;
- The prototype needed to be complex enough to warrant adaptation;
- The prototype needed to be feasible given the currently available mobile technology and needed to be able to run on mobile devices available on the market today;
- The prototype needed to be representative of the MMV domain and therefore needed to support major IV tasks (Section 2.2) and major MMV tasks (Section 2.5.2); and
- A topical and interesting application domain was also required.

It was decided to implement an application which allows users to visualise location-tagged multimedia captured using their mobile phone in a map-based view. Location-tagging of multimedia is proving a popular trend. This is being driven by technology advances in mobile devices. The quality of cameras available on mobile devices is rapidly improving. Advanced functionality (including ISO settings and scene modes) is becoming more common, making high-end mobile camera phones available today similar in terms of capabilities to entry level cameras. Furthermore, devices with storage capacities of many gigabytes are now available, providing the necessary capacity to store large collections of high quality multimedia.

Location has been shown to be one of the most memorable variables for users trying to locate previously captured photos (Naaman *et al.* 2004). Map-based visualisations of media collections therefore provide the potential to assist users in locating media items. Visualisation of media collections in this way also provides an interesting means of visualising media items captured while travelling. Users may not remember exactly where a photo was taken, but the ability to add location tags to photos at the time of capture provides interesting possibilities for reliving their travels and relating their experiences to others. Such an application therefore takes advantage of the personal nature of mobile phones.

Location-tagging and map-based visualisation of photo collections has emerged as a popular trend on the Internet. Websites such as Flickr (2008) allow users to upload location-tagged photos. Users can browse photos shared by others in a variety of ways, including a map view, as shown in Figure 5.1. Location-tagging capabilities are also becoming increasingly common in the high-end section of the mobile phone market. The Apple iPhone 3G provides integrated location-tagging capabilities (Apple 2008). Recent model Nokia N-series mobile phones now also integrate location-tagging into the standard camera application (Nokia 2008a). This functionality is also integrated with the Nokia Maps software available on these devices, allowing location-tagged photos to be visualised in a map view in this separate application. These devices make use of built-in GPS receivers in order to embed location information in the Exchangeable Image File Format (EXIF) data of the photos.

Several research systems have been developed to visualise photo collections on mobile devices. Many of these have incorporated context variables, including time and location, to allow users to browse their photo collections according to different dimensions (Harada *et al.* 2004; Pauty *et al.* 2005). Zurfer, a mobile Flickr client, was also recently developed to allow users to browse both their own photos and those of their friends, organised into channels (based on spatial, topic or social dimensions) (Naaman *et al.* 2008). Existing research systems have, however, rarely made use of maps.

The domain selected is therefore (as evidenced by the existence of similar systems) feasible given the currently available technology. It also represents a popular current trend in mobile computing. In the following section, the functional requirements of the prototype are presented in order to demonstrate that the remaining criteria are satisfied.

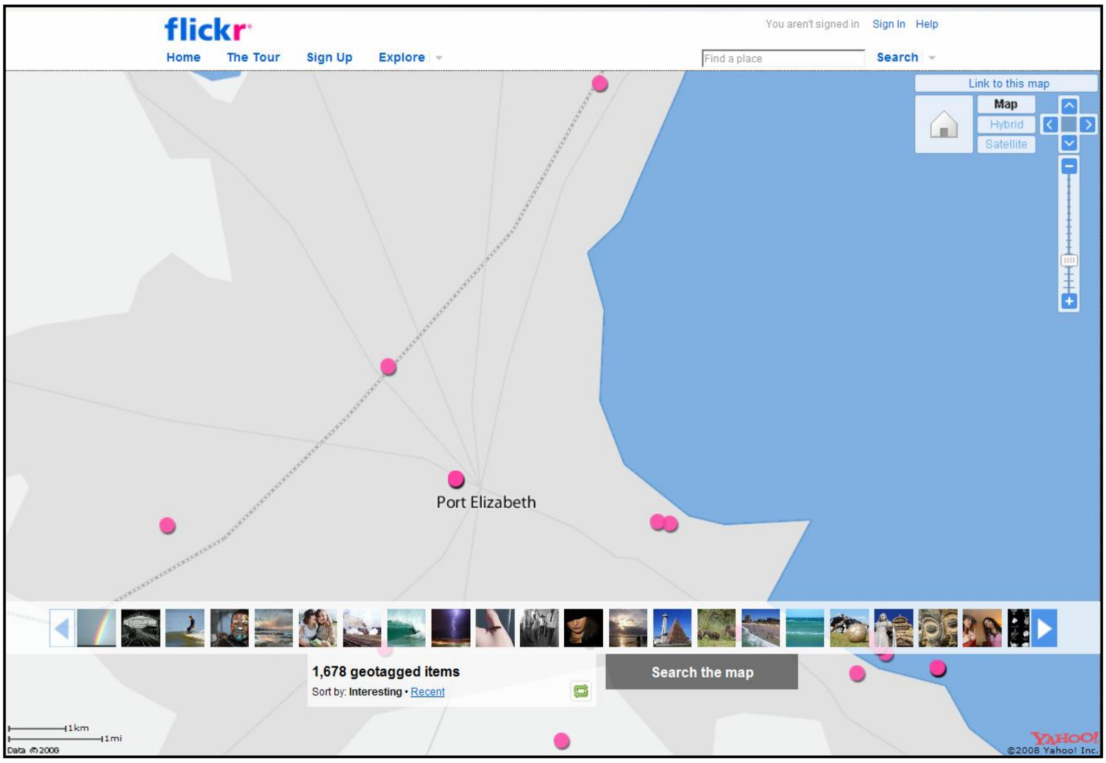

Figure 5.1: Location-tagged photos on Flickr. The pink circles denote the locations of photos.

5.3 Functional Requirements

The functional requirements for MediaMaps were determined by looking at extant systems in the application domain, including websites which allow users to visualise location-tagged photos, as well as commercial and research systems. Potential users were consulted to determine their requirements for such a system. This process produced the following functional requirements which were implemented in MediaMaps:

- Capturing and location-tagging of photos, videos and sound recordings;
- Organisation of multimedia into collections, based on time and location;

- Map-based visualisation of media collections and individual items (Figure 5.2), allowing users to zoom and pan, view details on demand, as well as filter the time period visualised;
- Animated map-based visualisation of individual items, allowing users to view media items in chronological order, showing where these media items were captured;
- List-based browsing of media collections;
- Searching of collections based on time and location; and
- Saving and loading of views.

Figure 5.2: Screenshot of map view in MediaMaps, showing media collections visualised

The map-based visualisation includes an overview function, which allows users to switch from the standard, detailed view to a zoomed-out overview by holding down a button. When the button is released, the view returns to the detailed view. A rectangular region shown in the overview helps the user identify how the current detailed view fits into the broader context (Figure 5.3). This is intended to help address the problem often encountered in MMV systems, where users lose track of how the current view fits into the global context (Section 2.6).

Figure 5.3: Overview function in MediaMaps

The above list shows that MediaMaps supports five of the seven information visualisation tasks (Section 2.2). The map view supports overview, zoom, filter and details on demand. Extract is supported through the saving and loading of views and associated parameters. Relate is partially supported through the use of collections and the animated view showing the temporal relationship between media items. MediaMaps also supports the orientation/localisation, identification, searching and (if media items are considered as events) event check tasks identified by Reichenbacher as typical MMV tasks (Section 2.5.2). MediaMaps can therefore be considered as representative of the domain of MMV. MediaMaps incorporates MMV as a central part of its functionality and is complex enough to potentially benefit from adaptation. MediaMaps therefore matches all of the criteria for a proof-of-concept prototype of the Proteus Model outlined in Section 5.2.

The implementation tools and development methodology used in the implementation of MediaMaps are now briefly described.

5.4 Implementation Tools

Java Micro Edition (Java ME) was selected as the implementation language. The prototype was developed to target mobile devices without touch screens. The Symbian operating system dominates this sector of the market and the smart phone market in general (Gartner 2008). This leaves a choice of two major implementation languages, namely Symbian C++ and Java. Java was selected because of its cross platform support, superior development tools and the availability of easy to use multimedia, location and file connection APIs. NetBeans 6.1 was used as the development environment, with various Software Development Kits

(SDKs) integrated in order to produce installers for different phone models. A Nokia N95 and a Nokia N95 8GB were primarily used for testing purposes, both of which require the Series 60 3rd Edition Feature Pack 1 SDK. Nokia currently accounts for almost 50% of sales in the current smart phone market (Gartner 2008).

5.5 Development Methodology

The development of MediaMaps followed a model-based design approach. The design and implementation of the system was based on the Proteus Model (Chapter 4), as the system was implemented as a proof-of-concept of this model.

A process of iterative refinement through evaluation of successive versions of the system was followed in the design of the user interface. Users were involved in order to analyse and improve on initial versions of the prototype. An iterative approach is generally followed in the development of IUIs (and thus AUIs) consisting of the following steps (Ehlert 2003):

- Analysis;
- Development and implementation of prototype interface techniques and metaphors;
- Evaluation; and
- Adjustments are made based on the evaluation results.

5.6 Architecture

Figure 5.4 shows the implementation of MediaMaps in terms of the Proteus Model architecture. The most noticeable change is that the additional data which is visualised is stored on the device itself and as a result, there is no link via the Internet to access an additional data source. All the other components were implemented as specified in Chapter 4. Components not involved in the adaptation process are discussed first.

5.7 Non-AUI Components

MediaMaps implements all the components of the Proteus Model. The non-adaptive components will be described first in order to outline the basic system functionality in more detail before the different forms of adaptation are described.

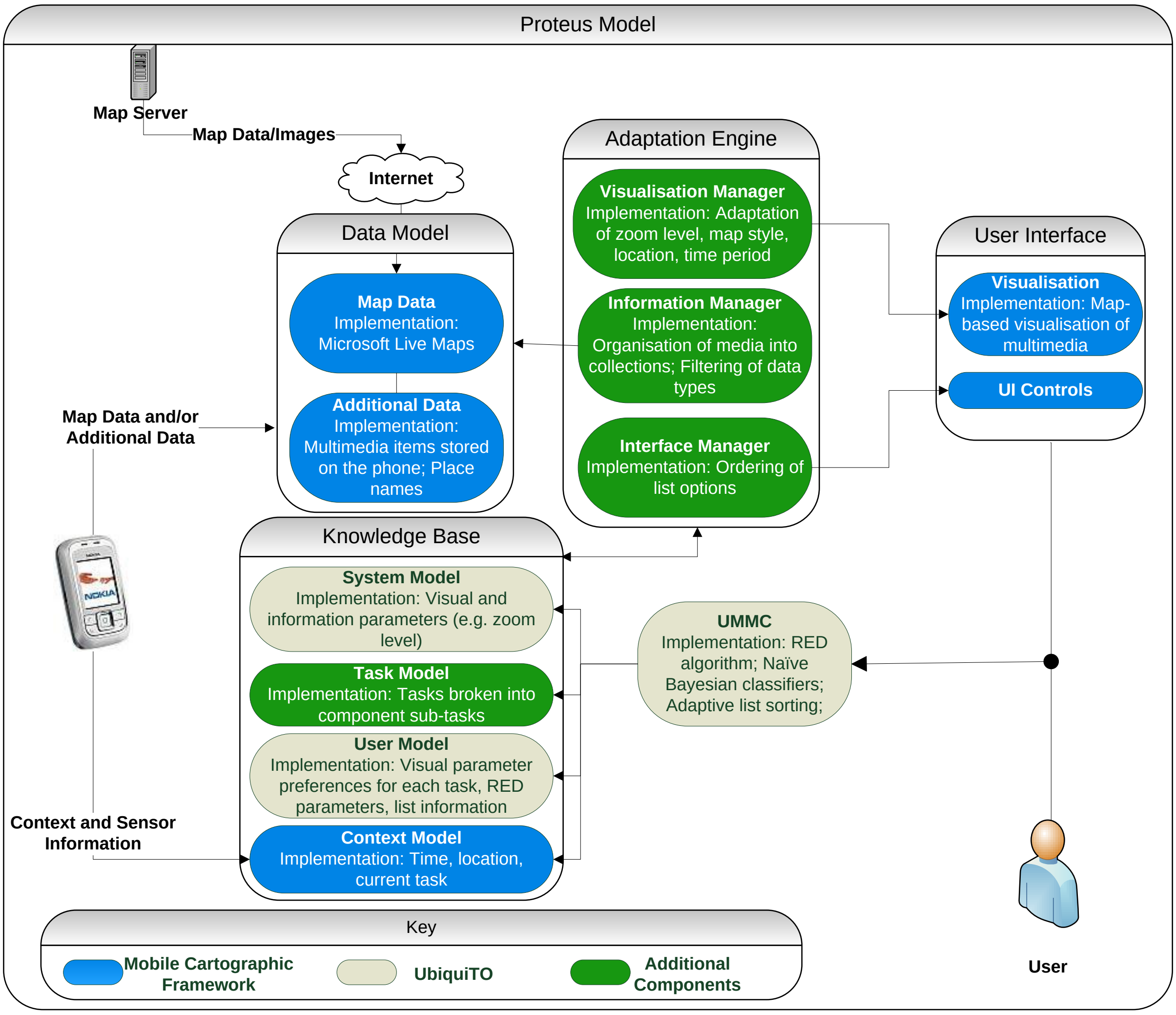

Figure 5.4: Implementation of the Proteus Model in MediaMaps

5.7.1 Data Model

The Data Model consists of both Map Data and Additional Data (Figure 5.3). The additional data refers to any data which is visualised in addition to the maps.

5.7.1.1 Map Data

Maps used in MMV systems are either rendered from raw map data, or downloaded as image tiles from the Internet. Rendering maps from raw data provides the benefit of greater levels of control over the rendering process. For example, only certain map features can be rendered or the level of detail at which to draw the maps can be adapted. There are also several disadvantages. Map data can consume a large amount of disk space. Large volumes of data are necessary in order to accurately render detailed cartographic representations of an area. Rendering of maps can also be slow on resource constrained mobile devices. Furthermore, if maps need to be updated, the corresponding data needs to be updated in order to reflect these changes. The alternative approach is to download pre-rendered maps as image tiles via a variety of services that are available on the Internet. Google[8], Yahoo[9] and Microsoft[10] all provide map services that are used by other websites. Other initiatives, such as OpenStreetMap[11], allow individual users to contribute to online mapping services. Using such map services means that the developer has little control over the appearance of the maps. Maps are merely requested by specifying a location and zoom-level (and sometimes also a tile size). There is also potentially a cost involved because of network data transfer charges.

Raster maps do, however, provide several advantages. The use of pre-rendered maps is generally more efficient, as the map tiles merely have to be rendered as images and no computation is required in order to generate the maps. For the purposes of this research, our focus is on adapting MMV systems in general, rather than just the maps themselves. Furthermore, catering for all possible user locations would require a huge amount of map data to be stored. Once the data is obtained, this still needs to be translated into graphical form. This can be a difficult task on mobile devices with limited graphical APIs. As a result it was decided to make use of pre-rendered map images downloaded via the Internet.

Maps were downloaded as raster map tile images from Microsoft Live Maps (Microsoft 2008). This map service provides reasonably detailed maps of South Africa. These maps are also available in three variants, namely road maps, satellite photographs and a hybrid of the two where place names are superimposed on satellite photographs. These map tiles are recursively organised into quadrants and can be downloaded from tile servers via the Internet by requesting the correct file name. Quadrant numbers are appended onto the filenames as the zoom-level increases. For example, the file name `r312.png` indicates that tile requested is of the road map variety (signified by the “r” in front), and that it goes down three zoom levels, first into quadrant three, then quadrant one and then quadrant two.

---

[8] http://maps.google.com

[9] http://maps.yahoo.com

[10] http://maps.live.com

[11] http://www.openstreetmap.org

Source code for a mobile version of Microsoft Virtual Earth (Fuller 2006) was used as a starting point for the implementation of MediaMaps. This code was, however, written in C# and needed to be translated to Java ME. Based on this original implementation, a three-level system for accessing map tiles was incorporated into MediaMaps, ensuring that map tiles do not have to be downloaded repeatedly. When a map tile needs to be accessed, the system searches for the image in RAM. If it is available, the tile is rendered, otherwise the system looks for the tile in the phone's secondary storage. If the tile is not found in secondary storage, it is downloaded via the Internet. All downloaded tiles are saved to disk. Several tiles are also loaded into RAM, in order to ensure fast access to recently accessed map images. Different cache sizes were tested, with the final implementation caching ten tiles in memory. Multi-threading is used to ensure that the user is still able to interact with the map while map tiles are being downloaded. Requested map tiles are added to a list that a separate thread downloads while the user continues to browse the map. Placeholder tiles are displayed while maps are being downloaded. MediaMaps also detects when an Internet connection is not available and provides appropriate error messages.

5.7.1.2 Additional Data

The "Additional Data" in the Proteus Model refers to data over and above the maps themselves that plays a part in the visualisation process. In the case of MediaMaps, this is the individual media items and collections of media items captured using the system and stored on the mobile device. Collections are stored in MediaMaps using XML (Figure 5.5). Each collection has an associated XML file. Each media item has an associated filename, latitude, longitude, date/time stamp and location label. Location information was initially embedded in the EXIF data of photographs, but this was later changed to ensure a consistent approach for all media types. Captured media are stored in sub-folders of the standard media folders on the mobile device.

Place names of over 1000 cities in over 120 countries are also stored in the system in order to provide meaningful place name labels. These are organised into a hash table based on country name and then indexed based on latitude and longitude in order to ensure efficient labelling of media items. The smallest distance between the user's location and candidate place names is determined and the associated place name assigned to the media item in question.

```
- <sound>
    <filename>Sea</filename>
    <latitude>-34.045715078564</latitude>
    <longitude>24.927925329874</longitude>
    <modified>1221302313732</modified>
    <location>Port Elizabeth, South Africa</location>
  </sound>
- <photo>
    <filename>Beach</filename>
    <latitude>-34.045206381053</latitude>
    <longitude>24.928138816867</longitude>
    <modified>1221302943892</modified>
    <location>Port Elizabeth, South Africa</location>
  </photo>
+ <photo>
- <video>
    <filename>Jbay</filename>
    <latitude>-34.047402355034</latitude>
    <longitude>24.926949341438</longitude>
    <modified>1221303156864</modified>
    <location>Port Elizabeth, South Africa</location>
  </video>
</Media>
```


Figure 5.5: Extract from media collection XML file in MediaMaps

5.7.2 User Interface

In the Proteus Model, a distinction is made between the visualisations and the supporting user interface. In terms of implementation, this may not always be clear as shortcuts and interface controls may be built into the map visualisations. The visualisations provided in MediaMaps are first described, followed by a discussion of the user interface (UI) controls.

5.7.2.1 Visualisations

On start-up, MediaMaps provides users with a map view, showing an overview of the location of the collections that they have captured. Collections are denoted by icons superimposed on the map at the location of the first media item in a collection. A red/green colour spectrum is used to visualise time. Colour is an important tool for encoding information in visualisation systems and has previously been used to encode a variety of variables, including age (Heer 2004). A relative time scale between the newest and oldest item determines the colour of the border drawn around the collection icons. Newer collections are at the green end of the scale, with older collections denoted by a red colour. Interpolation is used to determine the border colour for collections falling between the two extremes of the time scale. Figure 5.6 shows the standard map view in MediaMaps, showing media collections denoted by icons on the map with coloured borders used to denote the age of the collections.

Figure 5.6: Visualisation of media collections in MediaMaps (landscape view)

Users are also able to view the individual media items within a collection by selecting the “View on Map” option from the list of collections (or while viewing an individual item). Individual items are visualised similarly to the collections, with different icons used to indicate photos (camera icon), videos (video camera) and sound recordings (speaker). Figure 5.7 shows an example of a visualisation showing individual media items in MediaMaps.

Figure 5.7: Individual media items visualised in MediaMaps

MediaMaps also provides an animated view that allows users to view an animated visualisation of media items in a collection. “Play”, “Pause”, “Next” and “Previous” controls are provided, allowing users to view an animation of the location of different media items within a collection and to skip between items. This is intended to help users relive their travels and allow them to see and communicate to others where and when individual media items were captured. A screenshot of this view is shown in Figure 5.8.

Figure 5.8: Animated view of a photo collection in MediaMaps

A similar type of visualisation was employed in a desktop environment by Aris *et al.* (2004) in their system which included a "Trip Replay" visualisation.

5.7.2.2 User Interface Controls

MediaMaps uses a predominantly menu-driven interface. The main menu is shown in Figure 5.9 and provides the following options:

- *Capture:* Allows the user to capture different types of media (photos, videos or sound).
- *Browse:* Allows the user to browse collections in either a map view display or a list view display.
- *View:* Allows the user to switch to animation view, save and load views, as well as to zoom in and out of the current map view.
- *Search:* Allows the user to search by date or place name.
- *Controls:* Provides the user with an overview of the system controls and shortcuts.
- *Settings:* Allows the user to customise the map style (road, satellite photograph or hybrid), filter the media types which are displayed and toggle adaptations on or off.

Figure 5.9: MediaMaps main menu

The user is able to interact with the map view in a variety of ways. Standard zooming and panning controls are provided. The user is also able to select collections or individual media items in order to view (or hear, in the case of sound) the media item or items and view associated information (details on demand). The user is also able to filter the time period currently visualised using the keypad. Tool-tip type information is provided at the top of the screen when the selection pointer moves over an icon on the map.

A calendar metaphor is used to support searching by date. Small icons are placed on the relevant days on the calendar to indicate what types of media were captured on that day. The user is able to select either an individual day, or a date range. Media collections which occur on that day (or within the selected date range) are listed and the user is able to select one for viewing. The search by date screen is shown in Figure 5.10. Efficient searching is supported through the use of indexing. Media items and collections are indexed based on date and location at system start-up. A Binary Search Tree (BST) is used for the date index, while a hash table is used for the location index. A BST is preferred for the date index, as ordering information is important, since searches can be performed using a date range (unlike the place name index). Furthermore, as new media items are captured, these need to be added to the index, while maintaining the sorted order. A custom BST class was created with methods for efficiently accessing media or collections falling in a specified date range.

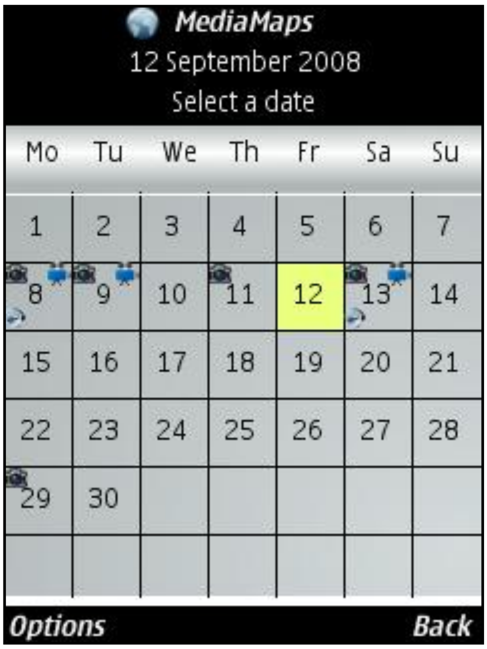

Figure 5.10: Search by date screen in MediaMaps

Once a particular media collection is selected, it can either be viewed on the map (showing the individual items), or browsed using a more traditional interface such as that shown in Figure 5.11 and Figure 5.12. Collections are categorised by media types. All the items of a particular media type can then be browsed using the left and right arrows.

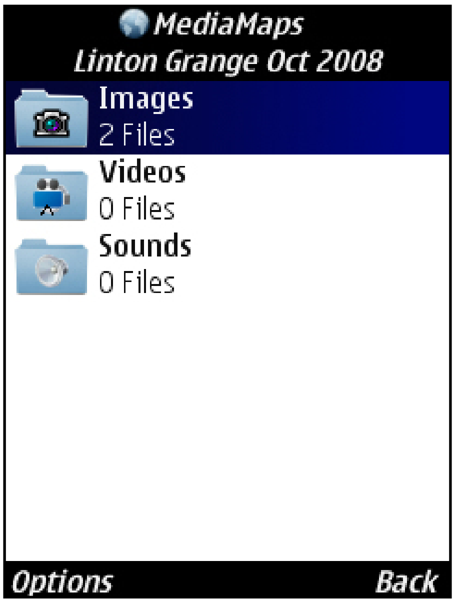

Figure 5.11: Browsing a media collection in MediaMaps

Figure 5.12: Browsing a particular media type within a collection

The discussion of the non-adaptive components above illustrates the general functionality provided by MediaMaps. The different forms of adaptation facilitated by MediaMaps are described in the following section.

5.8 Adaptation

MediaMaps implements all three forms of adaptation facilitated by the Proteus Model. Each of these forms of adaptation is discussed and motivated in the following sections. More detail is provided later in the chapter regarding how these adaptations utilise the components in the Proteus Model.

5.8.1 Visualisation Adaptation

A wide variety of visual parameters can be adapted in MMV systems, many of which have been targeted in existing systems and research (Section 3.10.2). The use of raster maps in MediaMaps limited the types of visualisation adaptation that were possible (for example, it was not possible to only paint certain map features, as the maps downloaded from Microsoft Live Maps are not customisable). Nevertheless, several possibilities for visualisation adaptation remained. One of the most significant problems of standard zooming and panning techniques (as used by MediaMaps) is that the user often needs to perform many zooming and panning operations to find information of interest (Section 2.6.2). To combat this, the map-based media visualisations in MediaMaps are adapted according to previous user behaviour. This was done in order to save users from having to perform the same view customisation operations every time they view a particular collection. Several simple parameters were selected to be adapted according to the user's behaviour. These were:

- The zoom-level (map scale) at which the map is visualised;
- The latitude and longitude at which the map is centred;
- The time period currently visualised; and
- The map style in use (road map, satellite photograph or hybrid of the two).

Naïve Bayesian classifiers (Section 3.8.2) were used for the discrete-valued variables, in order to adapt the visualisations according to user behaviour. These Naïve Bayesian classifiers are used to determine user preferences, based on past user behaviour. The values of several discrete-valued input attributes $\{A_1, A_2, \ldots, A_k\}$ are used to predict the value of another discrete-valued variable (C). The classifiers are used to determine the value c such that $P(C = c \mid A_1 = a_1 \wedge A_2 = a_2 \wedge \ldots \wedge A_k = a_k)$ is maximised.

According to Bayes' Theorem, this value is equal to:

$$\frac{P\ (A_1 = a_1 \wedge A_2 = a_2 \wedge \ldots \wedge A_k = a_k \mid C = c).P(C = c)}{P(A_1 = a_1 \wedge A_2 = a_2 \wedge \ldots \wedge A_k = a_k)}$$

For example, if the user's preferred zoom level needs to be determined (given the currently visualised collection and mode) the formula would look as follows:

$$\frac{P(Collection = \text{My Collection} \wedge Mode = \text{static} \mid Zoom\ Level = \text{i}).P(Zoom\ Level = \text{i})}{P(Collection = \text{My Collection} \wedge Mode = \text{static})}$$

Probabilities are calculated for all possible zoom levels and the maximum value returned. Naïve Bayesian classifiers require that all attributes used as input to the classifier be independent of one another. While this assumption is usually violated, classifiers have been shown to achieve accuracy levels comparable with more complex methods. Furthermore, the training time required is linear to the number of input variables multiplied by the number of training examples (Elkan 1997). Learning can therefore take place fairly quickly, with minimal resource requirements. As a result, Naïve Bayesian classifiers are well suited to a mobile environment and have been used successfully in a variety of existing systems (Nurmi and Hassinen 2007; Boström *et al.* 2008a). For the continuous-valued variables (time and location), a simple averaging calculation across the training examples was used.

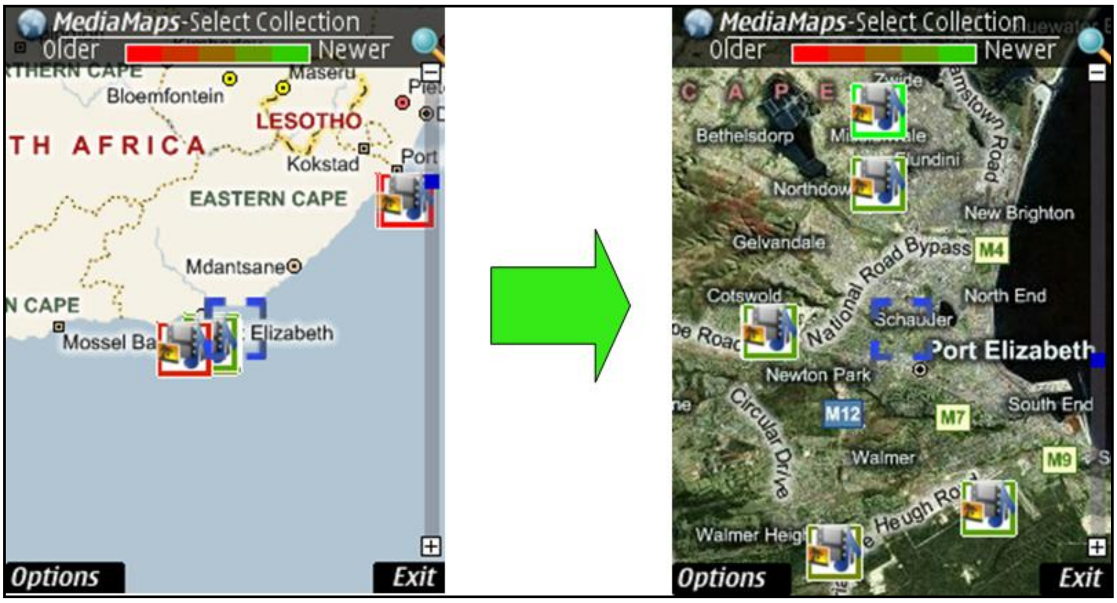

Figure 5.13: Visualisation adaptation. The same data set is shown before and after adaptation.

User interaction data is observed and logged throughout system use. The values for the above parameters, as well as the collection and mode (static or animated) are recorded, in order to be used as training examples for the Naïve Bayesian classifiers. An example of Visualisation Adaptation is shown in Figure 5.13. The same collection is shown before adaptation (left) and after adaptation (right). The map type (road map to hybrid), zoom level and location have all been adapted according to previously observed user behaviour.

5.8.2 Information Adaptation

Section 3.10.2 identified several types of information adaptation possible in MMV systems. The selection and grouping of information were considered to be of particular importance for MediaMaps. Clutter has been identified as a significant problem in MMV systems (Section 2.5.3). As a result, the filtering of information according to user preferences is likely to be a useful adaptation. The awkward interaction techniques currently available, combined with the ever increasing volumes of media items being stored on mobile devices, create a need for better ways of organising and retrieving multimedia on such devices. Furthermore, if a user's entire media collection (potentially containing thousands of photos) is to be visualised in a map-based view, screen clutter is likely to be particularly severe. As a result, information adaptation was implemented in MediaMaps in two forms. Firstly, the data types selected to be visualised are filtered according to user behaviour. This is done through the use of a Naïve

Bayesian classifier, similarly to the visualisation adaptation discussed above. Secondly, media items are sorted into collections based on time and location, using an algorithm developed by Chen and Chen (2007) that adapts according to the user's individual media capturing behaviour (Section 3.9.2). The sorting of media items into collections helps to minimise the clutter problem that would otherwise result if a user's entire media collection was visualised in a single view.

The real-time event detection (RED) algorithm developed by Chen and Chen follows on a large amount of related work that attempts to organise photo collections based on a variety of variables, including time and location (Loui and Savakis 2000; Platt *et al.* 2003; Naaman *et al.* 2004; Cooper *et al.* 2005). Two approaches are evident in existing research. In the first, more common approach, the photo capturing phase is separated from the photo organising phase, with complex clustering algorithms applied to photo collections after they are captured. In the second approach, a decision regarding the sorting of a photo is deferred until several subsequent photos have been captured. These approaches are undesirable in a mobile environment, as they are resource intensive and may result in significant delays before photos are sorted. Unlike the majority of existing research, the RED algorithm provides for the sorting of photos incrementally and in real-time (as they are captured). Furthermore, unlike many other existing algorithms, the RED algorithm was specifically designed for use on mobile devices and is therefore designed to operate in resource-constrained environments. Photos can be classified as they are captured, allowing for immediate user feedback. This also means that the algorithm has the capability to adapt, with user responses to system recommendations used to improve the accuracy of the algorithm.

The RED algorithm attempts to match the user's photo taking behaviour to a Poisson statistical process. A Poisson process has the following four properties:

- Increments are independent of one another;
- Events cannot occur simultaneously;
- It is stationary (the experimental outcome of an observation is not affected by its start time); and
- $N(0) = 0$ where $N(t)$ is the number of events occurring in the time interval $[0,t]$.

The first two properties are naturally satisfied by an individual user's photo taking behaviour. Some special measures are required to ensure that the other two properties are satisfied. To ensure that the stationary property is satisfied, photos are only regarded as part of the Poisson process when the user is considered to be active in capturing multimedia. If a long time passes without a photo being captured, the user is considered to be inactive. To satisfy the final property, the origin of the time period under consideration is set to the time stamp of the first photo captured as part of each new event. The probability function of the Poisson process is:

$$\Pr\{N(t) = k\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

where λ is the arrival rate (the number of photos the user takes in an hour), t is the number of hours that have passed since the start of the current event and k is the number of photos that have been captured in t hours.

Figure 5.14 shows the Poisson process according to which the user's photo taking behaviour is modelled in the RED algorithm. Three regions are shown, delineated by lower and upper confidence limits at 0.2 and 0.8 respectively (these are the default values for these parameters recommended by Chen and Chen). The time gap between the current photo and the previously captured photo is used to determine in which of the three regions the current photo belongs. The three regions and their respective meanings are:

- R_o: A photo is taken a short time after the previous photo and is classified as belonging to the current event.
- R_l: A photo falls into the intermediate zone between a new event and an old event. Location information is incorporated to help classify the photo.
- R_n: A photo is taken a long time after the previous photo and is classified as belonging to a new event.

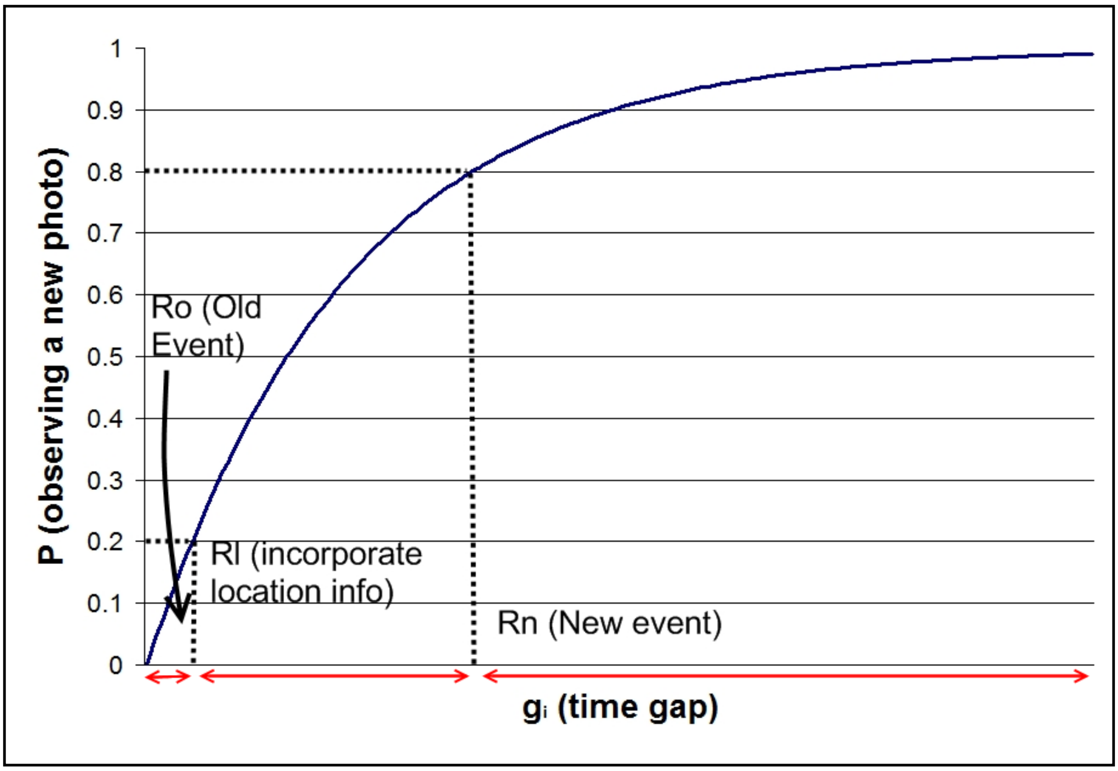

Figure 5.14: Modelling user photo capturing behaviour according to a Poisson process. Adapted from (Chen and Chen 2007)

If photos fall into the intermediate zone between new and old events, the distance between the current photo and the previously captured photo is taken into account in order to classify the photo. The following formula is used to decide where the newly captured photo belongs to the current collection, or to a new collection:

$$d_i < \mu_d + \left(1 + \frac{g_i}{\bar{g}}\right) x\ \sigma_d$$

Where:

d_i = the distance between the current photo and the previous photo;

μ_d = the mean distance between photos;

σ_d = the standard deviation of the distances between photos;

g_i = the time difference between the capture times of the current photo and the previous photo; and

$\bar{g}$ = the mean time difference between photo timestamps.

If the above condition is satisfied, then the photo is identified as belonging to the current collection. Otherwise, it is identified as belonging to a new collection. This formula is used to determine whether the current distance gap is significantly larger than the mean distance between photos. As can be seen from the formula, the time gap (g_i) is also taken into account, compensating for the fact that the distance (d_i) is likely to be larger, as g_i increases. Photos that are taken within twenty metres of each other are considered duplicate photos of the same area. Only unduplicated photos are used in calculating μ_d and σ_d, ensuring that these values are not skewed by multiple photos being taken at the same location. Default parameter values of 0.5 for λ (the arrival rate of new photos per hour), and 500m for μ_d and σ_d are suggested by Chen and Chen. These parameters are updated as photos are taken, using the update method shown in Figure 5.15 (where D refers to the decision made regarding whether the photo belongs to the current collection or a new collection).

Subroutine **UpdateParameters** (D)

if (D = *New*) then
{
  λ is updated as the average arrival rate of the most recent *n* events;
  μ_d and σ_d are set to their initial values;
}
else
{
  $$\mu_d = \frac{1}{n}\sum_{i=1}^{n} d_i$$
  $$\sigma_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \mu_d)^2}$$
}

Figure 5.15: Update method for updating parameters in the RED algorithm (Chen and Chen 2007)

In MediaMaps, the RED algorithm is extended to include photos, videos and sound recordings (collectively referred to as media items). Despite the fact that the algorithm was developed for use on mobile devices, the original implementation by Chen and Chen (2007) was implemented and evaluated on a desktop system. This research therefore represents the first implementation of the RED algorithm on actual mobile devices.

System recommendations are presented to the user, who can choose to either accept or reject the system's recommendation. If a recommendation is accepted, the media item is added to a collection and the algorithm parameters remain unchanged. If a recommendation is rejected, the system must modify its behaviour in order to adapt to the user's behaviour. These modifications depend on the region where the error occurred.

- R_o: If the system believes a media item belongs in the current collection, but the item actually belongs to a new collection, the arrival rate needs to be increased (a new event has occurred sooner than expected given the current parameters). The algorithm updates the arrival rate using the `UpdateParameters` method (Figure 5.15) as the average of the last n events. A value of n=5 was used (the same value used by Chen and Chen).

- R_l: If the error occurs in the intermediate area where location information is incorporated in the decision-making process, the distance parameters, μ_d and σ_d are updated using the `UpdateParameters` method (Figure 5.15). The update depends on whether the media item belongs in the current collection (if the media item is unduplicated, μ_d and σ_d are recalculated) or in a new collection (μ_d and σ_d are set to the initial values).

- R_n: If the system believes a media item belongs in a new collection, but the item actually belongs to the current collection, the arrival rate needs to be reduced. This is not catered for in the `UpdateParameters` method. In this case, the arrival rate is reduced by removing the item with the shortest time gap from the last n items used to calculate the arrival rate.

A recommendation is presented to the user immediately after a new media item is captured in MediaMaps. Figure 5.16 shows a sequence of screenshots detailing this process. The user captures a photo and then manually location-tags it (the user's GPS position was unavailable). The system analyses the currently captured item and decides that it falls into the region R_n (new collection) and presents this recommendation to the user. The user is then able to accept or reject this recommendation. Suggested collection names are generated by MediaMaps for new collections which combine the place name corresponding to the user's location and the date.

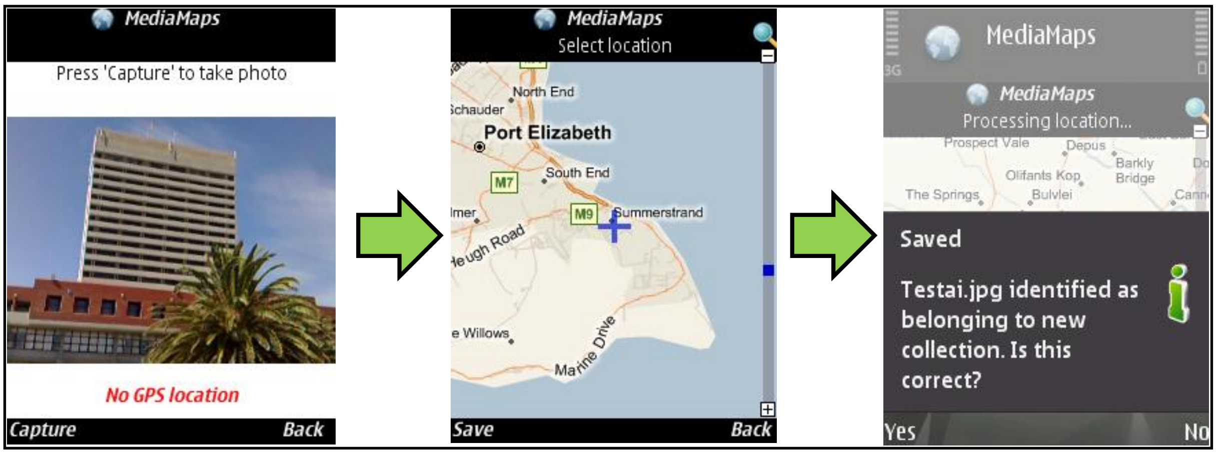

Figure 5.16: Recommendations in MediaMaps regarding the sorting of media items

5.8.3 Interface Adaptation

Several possibilities for interface adaptation exist in MMV systems (Section 3.10.2). Adaptable parameters include the availability and ordering of shortcuts, the ordering of menu options and the input modality. MediaMaps is targeted at keypad-based mobile phones, with Nokia N-Series devices being the target platform. Applications on these phones are usually menu-driven. Shortcut icons are occasionally used, but in this case would obscure the already limited display area. As a result, the ordering of menus and lists of options was the most obvious choice for adaptation.

Interface adaptation is implemented in the form of list ordering in MediaMaps. MediaMaps has one main menu that provides access to major system functions and several short menus on other system screens. It was decided that adapting the ordering of options on these menus was unlikely to provide significant benefits and that adapting the main menu was likely to prevent users from becoming efficient in performing frequent system tasks. The limited length of these menus also means that re-ordering of options is unlikely to achieve significant benefits. There are, however, several lists of options in MediaMaps of significant length (and others which could grow to significant length over time). Evaluation of prototype versions of MediaMaps revealed that users spent significant time browsing these lists. Such lists include the list of media collections, list of cities and the list of saved views.

Several algorithms for menu/list ordering were considered. Findlater and McGrenere (2008) used an algorithm for ordering menu options and tested it on desktop screens and in a simulated small screen environment. They found that the benefits of menu ordering were potentially greater on small screen devices than on the desktop. Their algorithm (Figure 3.5) moves the most recently selected option and the two most frequently selected options to the top of menus/lists. This algorithm was implemented in MediaMaps in lists of options with one modification. Items were moved to the top, adaptive section, rather than duplicated in this section. Informal evaluation revealed that duplicating the list option resulted in user confusion, particularly if the list of options was short. If the same item appeared in the adaptive and non-adaptive portions of a short list of options, users were unsure which item to select.

An example of interface adaptation in MediaMaps is shown in Figure 5.17. The top section (in grey) contains the adaptive section of the list, including both the most recently selected option and the two most frequently selected options, in the same relative order (alphabetical) as they would appear in the list below. The grey shading is used to draw attention to the adaptive portion of the list and also to denote the fact that these items are ordered separately from the rest of the list.

Figure 5.17: Interface adaptation (list ordering) in MediaMaps

In Section 3.5, several problems and shortcomings of AUIs were identified. Some of the most significant problems related to the fact that AUIs do not support the principle of user control and freedom. Another significant problem relates to users not understanding the underlying logic of the adaptation process. In MediaMaps, the user retains full control over the adaptation process. The sorting of media items into collections makes use of recommendations, rather than autonomous system actions, ensuring that the user stays in control of the sorting process. Visualisation and interface adaptation can also be disabled by the user (Figure 5.18). In order to combat possible user confusion regarding the adaptation process, short explanations are also provided to the user in order to explain the logic behind all three forms of adaptation. MediaMaps therefore represents a hybrid of the Self-Adaptation and User-Controlled Self-Adaptation classes of AUIs (Section 3.3).

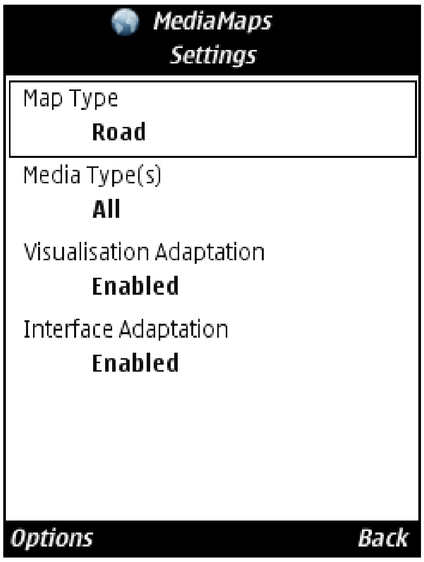

Figure 5.18: Settings screen showing options for disabling visualisation and interface adaptation

5.9 AUI Components

MediaMaps was developed primarily as a proof-of-concept of the Proteus Model. Components not involved in adaptation were described in Section 5.7. The previous section outlined the algorithms that were used to implement the three forms of adaptation facilitated by the Proteus Model. In this section, the data and interface implementations of the Proteus Model components which implement these three forms of adaptation are described.

5.9.1 The User Monitoring and Modelling Component (UMMC)

The UMMC is the first of the components involved in facilitating adaptation in the Proteus Model to be discussed. In Section 4.6.1 the role of the UMMC in observing, recording and modelling user interaction was described. In MediaMaps, these functions are split across two sub-components, namely the Monitoring Module and the User Modelling Component (UMC). The Monitoring Module is responsible for receiving interaction and context data (in the case of MediaMaps this is only location information). The location information is recorded in the Context Model, while the interaction data is recorded at system shutdown in XML files. Monitoring of location information can be deactivated, as this can deplete battery life significantly if left running continuously, given the power consumption of GPS receivers in mobile devices. Low-level interaction data is sent to the Monitoring Module through the use of event codes and the Task Model is used to identify the current user task, given recently observed interaction data. Once a task has been identified, the Context Model can be updated to store the user's current activity, so that the appropriate manager components can be invoked in order to perform adaptation. Simple User Model updates that do not require additional user modelling can also be performed directly from this component, such as updating adaptive lists and the RED algorithm parameters. For example, as soon as the user selects an option from an adaptive list, the User Model is updated so that this option is stored as the most recently selected item and its frequency of selection is also incremented.

Figure 5.19 shows an example of a single interaction record as captured by MediaMaps. These interaction records are used as training examples for the Naïve Bayesian classifiers used for visualisation adaptation (Section 5.8.1) and information adaptation (Section 5.8.2). The attributes recorded enable the adaptation of the zoom level, map style, location, media types and time period. Interaction data is eventually discarded on a first-in first-out (FIFO) basis in order to ensure that the system keeps pace with changing user preferences and that old interaction data does not skew what is stored in the User Model.


```
- <record>
    <time>1221217196767</time>
    <collection>Barcelona Aug 2008-1.xml</collection>
    <mode>0</mode>
    <zoomlevel>17</zoomlevel>
    <mapstyle>0</mapstyle>
    <latitude>41.39217551694313</latitude>
    <longitude>2.1479237079620366</longitude>
    <mediatypes>0</mediatypes>
    <startDate>0</startDate>
    <endDate>5</endDate>
  </record>
```


Figure 5.19: Extract from MediaMaps interaction log files

Interaction data related to interface adaptation (ordering of list options, Section 5.8.3) is recorded separately, with the most recent selection and frequency information for other list option selections recorded in order to facilitate this form of adaptation. Interaction data necessary to adapt the parameters of the RED algorithm used to sort media items into collections (Section 5.8.2) is stored in the collection files (Figure 5.5) and this data is used to update the algorithm parameters (where necessary).

The User Modelling Component class in MediaMaps is shown in Figure 5.20. It is responsible for processing interaction data in order to generate user preferences which are then stored in the User Model. The Context Model is also updated with the user's latest position where this is available via GPS or when selected via the manual location-tagging map. The work of the Bayesian classifiers is performed in the `bayesianSettings` method in the UMC class. The maximum probability value across all possible parameter values is calculated based on the user's interaction data and recorded as the user's preference (the `determineMaxIndex` method is used for this purpose). The Singleton design pattern (which ensures that only one instance of a class is created) is used throughout MediaMaps to ensure minimal memory footprint (note the use of `getInstance` methods in most major classes).

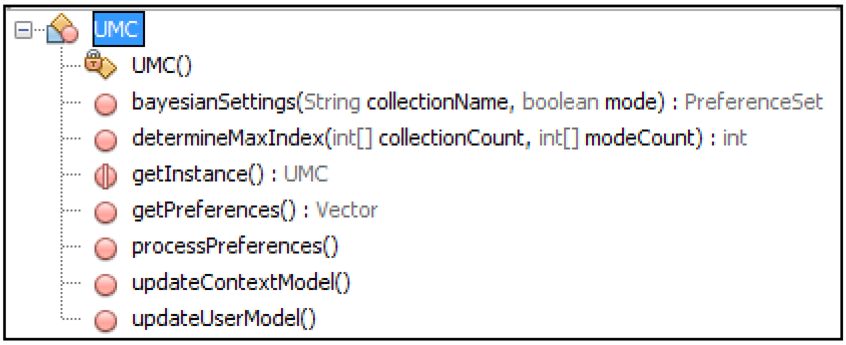

Figure 5.20: User Modelling implementation in MediaMaps

5.9.2 Knowledge Base

All four models contained in the Knowledge Base of the Proteus Model were implemented in MediaMaps and are discussed in the sections that follow.

5.9.2.1 System Model

The System Model in MediaMaps was implemented as a single class. This consists almost entirely of `get` and `set` methods for the different system parameters. The class also has a method for writing the model to disk at system shutdown. These parameters consist of:

- Visualisation parameters:
  - Zoom-level
  - Map Style
  - Location
  - Time period

- Information parameters:
    - Media types
    - RED algorithm arrival rate (λ), μ_d and σ_d

Interface parameters do not need to be stored in the System Model, as these values are only important when the relevant lists are being displayed and can only be adjusted by the Interface Manager. The most recently selected option and frequency count information for each list is stored in the User Model.

5.9.2.2 Task Model

The Task Model in MediaMaps contains different system tasks encoded as arrays of event codes, used to signify the different sub-tasks comprising an event. The Task Model allows the current user task to be identified based on observed interaction. A queue of the last few observed user interactions is maintained in order to resolve any ambiguities. The `getTask` method is used to return the relevant task and update the Context Model (if necessary) so that the user's current task is stored and relevant adaptations can be performed. Figure 5.21 shows the Task Model as implemented in MediaMaps.

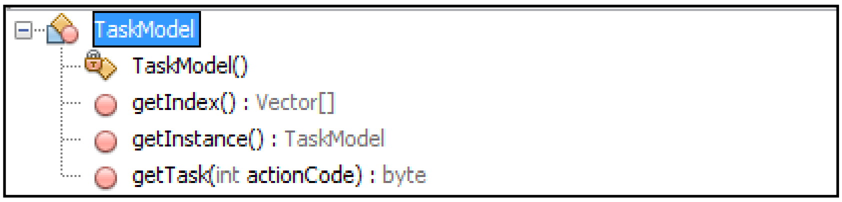

Figure 5.21: Task Model implementation in MediaMaps

5.9.2.3 User Model

The User Model in MediaMaps was implemented as a single class. The User Model stores the output of the UMMC. In MediaMaps, different information is stored for the different forms of adaptation, as follows:

- *Visualisation adaptation:* The User Model contains a hash table of user preferences for the different visualisation parameters for each collection (including the default map view) and mode (static or animated) combination. These are stored as Vectors of parameters.

- *Information adaptation:* The hash table containing visualisation parameters also contains the user's preferred media types. The parameters for the RED algorithm that have been learned based on user feedback (the arrival rate and mean and standard deviation distance between media items) are also stored separately in the User Model.

- *Interface adaptation:* In order to adapt the ordering of list options according to how frequently and how recently these options were selected, this information needs to be stored in the User Model. The UMMC updates this information as new list selections are made.

The User Model class in MediaMaps is shown in Figure 5.22. The class provides methods for retrieving the user preferences (different `getPreferences` methods are provided for the different forms of adaptation), loading the User Model at system startup (`loadModel`) and saving the model to disk at system shutdown (`updateModel`).

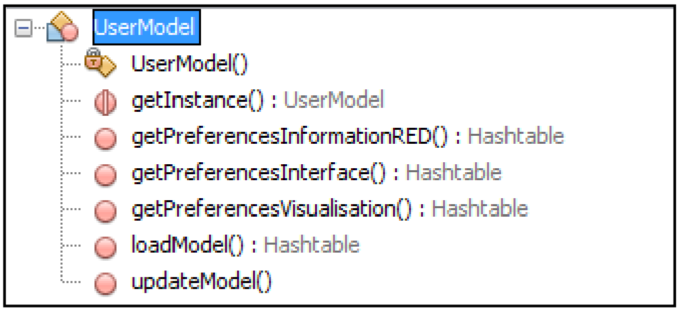

Figure 5.22: User Model implementation in MediaMaps

5.9.2.4 Context Model

The Context Model in MediaMaps stores only the user's current location (split into latitude and longitude) as well the user's current task. Time information is important in the RED algorithm used for information adaptation, but can easily be obtained by a call to the Java method `System.currentTimeMillis` and therefore does not need to be stored. The Context Model also contains methods to load the model (`loadModel`) at system start-up and to write the model to disk at system shutdown (`updateModel`) (Figure 5.23). Finally, the Context Model is responsible for invoking the different manager components. Different tasks have different managers associated with them. When the user's current task is updated in the Context Model, the task code is used to look up which of the manager components (Visualisation Manager, Information Manager and Interface Manager) are associated with that task and need to be invoked. The Context Model calls the `invoke` method of the appropriate manager components. The implementation of the different manager components is now discussed.

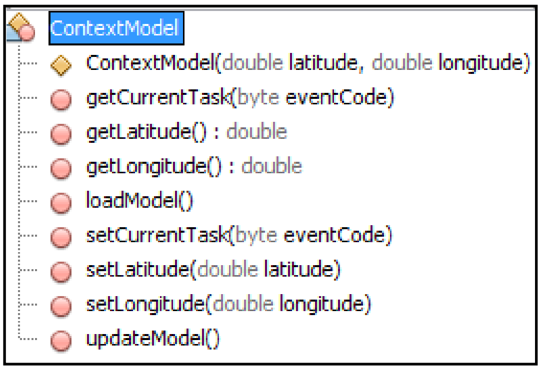

Figure 5.23: Context Model implementation in MediaMaps

5.9.3 Adaptation Engine

The Adaptation Engine in MediaMaps does not exist as a component on its own, but is used as a convenient collective term for referring to the different manager components. All three manager components in the Proteus Model were implemented in MediaMaps.

5.9.3.1 Visualisation Manager

The Visualisation Manager in MediaMaps (Figure 5.24) includes one important method (`invoke`) for performing adaptation. The Visualisation Manager retrieves the user's preferences from the User Model. The Visualisation Manager is only invoked when new visualisations are to be rendered. The current collection being visualised and the current mode (static or animated) are retrieved from the System Model. These parameters are combined to form a hash key to retrieve the appropriate preferences regarding visualisation parameters from the preference hash table stored in the User Model. The appropriate visualisation parameters are then adjusted and updated in the System Model before the visualisation is presented to the user.

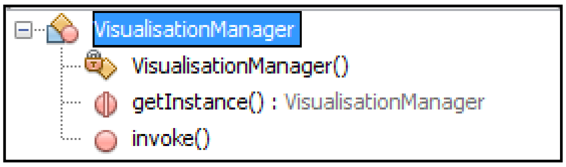

Figure 5.24: Visualisation Manager implementation in MediaMaps

5.9.3.2 Information Manager

Information Adaptation is performed in two ways in MediaMaps (Section 5.8.2). As a result, the Information Manager (Figure 5.25) provides two `invoke` methods, depending on the type of adaptation required. The first type of adaptation (where media types are filtered) functions similarly to the visualisation adaptation described above. Once invoked by the Context Model, the Information Manager retrieves the current collection and mode from the System Model. These are used to retrieve the appropriate user preference regarding media types from the hash table stored in the User Model. This parameter is then updated in the System Model and the visualisation is rendered.

The `invokeRED` method in the Information Manager is used to update the parameters of the RED algorithm used to provide recommendations to the user regarding the sorting of media items into collections (Section 5.8.2). This method is invoked when the user has captured a media item and rejected the sorting recommendation provided by MediaMaps. As a result, adaptation of the RED algorithm parameters is required. The Information Manager updates these parameters (arrival rate, the mean distance between media items and the standard deviation of these distances) in the System Model to ensure that media items are sorted according to user preferences.

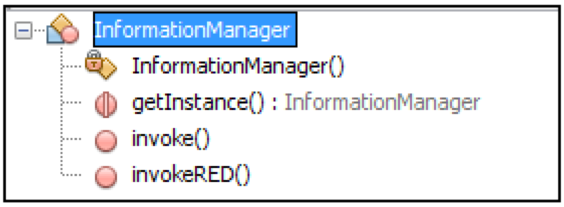

Figure 5.25: The Information Manager implementation in MediaMaps

5.9.3.3 Interface Manager

The Interface Manager is also activated through its `invoke` method. This method is invoked whenever an adaptive list is about to be rendered. All adaptive lists in MediaMaps implement an `AdaptiveList` abstract class. This abstract class provides a partial list implementation which ensures that list painting, ordering of list options and updating of the User Model occurs in a standard fashion and that this code does not need to be duplicated across multiple classes. The implementation of `AdaptiveList` invokes the Interface Manager when a list is to be rendered. The Interface Manager queries the User Model to determine the most recently selected option and the two most frequently selected options for that list. These options are then painted in the top section of the list (Figure 5.16) in the same relative order as they would normally appear in the list.

5.10 Conclusions

The map-based visualisation of location-tagged multimedia was identified as an appropriate domain for a proof-of-concept prototype MMV system based on the Proteus Model. This domain was selected as it incorporates MMV as a central component, was feasible given the currently available technology and represents a popular current trend in mobile computing.

Extant systems were reviewed and potential users were consulted in order to determine the functional requirements for the prototype MMV system, called MediaMaps. MediaMaps allows users to capture, location-tag, sort and visualise media items on their mobile phones. MediaMaps represents a typical MMV system, supporting many of the standard IV and MMV tasks.

MediaMaps was implemented using Java ME, which was identified as the appropriate implementation platform due to its cross-platform nature and the availability of excellent development tools and easy to use multimedia, location and file connection APIs. A model-based design approach was followed, with MediaMaps implementing all the components of the Proteus Model. Iterative development was identified as a popular approach to the development of AUIs and was used to successively improve the user interface design.

All three forms of adaptation facilitated by the Proteus Model are supported in MediaMaps. Information adaptation is supported through the filtering of media types and the automated sorting of media items into collections (based on time and location) in order to simplify media collection management and reduce screen clutter. The RED algorithm was identified as an appropriate algorithm for the sorting of media items into collections, due to its low resource requirements and adaptive nature. Naïve Bayesian classifiers are used to perform visualisation adaptation, adapting map-based visualisations of media collections according to user behaviour in order to prevent users from having to perform repetitive view customisations. Naïve Bayesian classifiers have low resource requirements and have previously been used successfully to adapt mobile systems. Interface adaptation is supported by providing users with easy access to their most recently and frequently selected options through the use of adaptive lists of options.

MediaMaps demonstrates the successful implementation of the Proteus Model and its individual components in an adaptive MMV system. In the following chapter, the evaluation of MediaMaps will be described. This evaluation was conducted in order to determine whether the adaptations implemented in MediaMaps provide any real benefits to users of the system.

Chapter 6: Evaluation

6.1 Introduction

In Chapter 4, the Proteus Model for adaptive MMV systems was presented. In Chapter 5, the implementation of a prototype MMV system, called MediaMaps, based on this model was discussed. In this chapter, the evaluation of MediaMaps is described. MediaMaps was evaluated in order to investigate the application of AUI techniques in an MMV system. A preliminary usability evaluation is described which was used to evaluate the effectiveness and satisfaction of the prototype. The results of an international field study of MediaMaps are presented. This field study evaluated the accuracy, usefulness and usability of the adaptations implemented in MediaMaps and the usability of the system.

This chapter begins with a brief review of the relevant literature regarding usability evaluation in the fields of mobile Human-Computer Interaction (HCI) and AUIs in order to guide the selection of appropriate evaluation techniques. The two evaluations of MediaMaps are described, including the objectives, instruments, participants and the test procedure used. The results are then presented and analysed.

6.2 Evaluation Methods

Existing literature from the fields of mobile HCI and AUIs was examined in order to determine the most appropriate evaluation methods, metrics and instruments to be used in the evaluation of MediaMaps.

6.2.1 Evaluating Mobile Applications

The goal of usability testing is to determine whether the product or system being tested is usable by the intended user population to achieve the tasks for which it was designed (Dumas and Redish 1999). Usability testing of desktop systems has traditionally been conducted in the controlled environment of usability laboratories (Preece *et al.* 2007). Mobile HCI research has adopted similar techniques. A study in 2003 showed that 71% of evaluations published in major mobile HCI and mobile computing conference proceedings and journals were laboratory experiments. Only 19% involved field studies examining the use of the application or mobile device in its actual context of use (Kjeldskov and Graham 2003). Mobile devices and applications can be used in a wide variety of contexts. Users can be on the move, or dividing their attention between their mobile device and another activity. Lighting and sound conditions can also vary, making the display potentially difficult to read outside or audio output difficult to hear. Laboratory evaluations of mobile applications cannot take such variables into account. While laboratory studies can still be useful in identifying specific usability problems, it is important that a mobile application be evaluated in its intended context of use if such variables are to be taken into consideration (Tamminen *et al.* 2004; Jones and Marsden 2005).

Field studies are not, however, appropriate for all situations. Kaikkonen *et al.* (2005) compared laboratory and field testing to evaluate a photo sharing application and found that similar problems were identified using both methods. They describe laboratory testing as being sufficient for identifying usability issues and improving user interaction. Their results do not extend to other types of usability testing. They also stressed the importance of conducting pilot tests before conducting costly field studies.

Field studies can take on various forms, including ethnographic studies (where the researcher is immersed to some extent in the test environment, observing the user interacting with the system), field experiments (where specific independent variables are manipulated) and field trials (where users use the system on their own and are required to provide feedback in one form or another) (Kjeldskov and Graham 2003; Preece *et al.* 2007). The appropriate technique depends on the goals of the particular evaluation. If the influence of a particular variable on some aspect of a system needs to be investigated, then field experiments are clearly the appropriate choice. If more general, qualitative feedback is required, then ethnographic studies or field trials are more appropriate.

Informal usability evaluations can also play an important part in the design phase, to ensure that systems are understandable and match the needs of their users (Jones and Marsden 2005; Jones *et al.* 2005). Heuristic evaluations can also be conducted that do not involve users at all, with experts evaluating the system against several heuristics, thereby attempting to identify potential usability problems.

A wide range of techniques have been employed to observe and record user interaction with mobile applications. It is often not just user actions that need to be recorded, but also observations regarding users' reactions to different aspects of the system under evaluation. The techniques employed are closely linked to the type of evaluation being conducted. In the controlled environment of the usability laboratory, video and audio recordings, note taking from direct observation and think-aloud techniques can be used. When conducting field studies, such approaches may not be feasible. Automated interaction logging and the use of questionnaires and interviews (either during or after system use) are usually favoured for field studies as they are less intrusive and allow the system to be used in a natural fashion (Kjeldskov and Graham 2003). During field studies, users are sometimes also required to complete diaries, detailing their experiences using the system and any problems they encountered. Reigelsberger and Nakhimovsky (2008) combined several different methods in their field trial of Google Maps for Mobile. They employed automated interaction logging, group briefing sessions and telephone interviews in their evaluation in order to counteract the shortcomings of the individual techniques. They did not use diary studies, perceiving these as being unreliable, as research has shown that users tend to make infrequent use of diaries.

6.2.2 Evaluating AUIs

The evaluation of AUIs has been identified as a fundamental part of developing such systems (Kobsa *et al.* 2001; Gena 2005). Despite this, existing AUI research has been criticised for containing a low proportion of reported evaluations, with less than 25% of publications in major AUI conference proceedings and journals reporting evaluations (Chin 2001; Masthoff 2002).

Evaluation of AUIs is problematic for the following reasons (Masthoff 2002):

- *Difficulty in attributing cause:* Adaptation is not the only variable that could influence the usability of a system. For example, if an AUI is evaluated and it is discovered that it helps users efficiently complete their tasks, it cannot conclusively be determined that adaptation was the root cause of this efficiency. Similarly, if the system is not efficient, this cannot be conclusively blamed on adaptation. One approach to combat this is to compare an adaptive version of the system with a non-adaptive version (Eklund and Sinclair 2000). A between-subjects study can then be conducted, with statistical tests used to test whether the adaptive system is a significant improvement over the non-adaptive system. If so, it would suggest that the adaptations improved the efficiency of the system. This technique is, however, not feasible in some circumstances. A non-adaptive version of a system is not always available, or cannot always easily be created without significant changes to the system having to be made. Adaptation is often an inherent part of the system and removing it may result in a system which does not optimally support user tasks (Höök 1999). Attributing cause can further be complicated by the fact that many AUIs incorporate more than one type of adaptation.

- *Difficulty in finding significant results due to variance:* One of the motivations for creating AUIs is to cater for differences in the user population (Section 3.4). As a result, there is often a large variance in user characteristics or behaviour. This can make it difficult for evaluations to produce statistically significant results.
- *Difficulty in defining the effectiveness of adaptation:* The effectiveness of an AUI can be interpreted in more than one way. This can be problematic, as the results can also be dependent on the point in time when the system is evaluated. Some adaptations have a noticeable effect immediately, while others may only be beneficial to users over time.
- *Difficulty in finding resources:* The evaluation of AUIs can be resource intensive, as many test participants are often required. If an adaptive system is to be compared to a non-adaptive system, then two user groups are required, as an order effect is likely if a within subjects design is used. Furthermore, to combat the variance in the underlying user groups, larger than usual participant groups are often required.

If adaptation is unsuccessful, it can also be difficult to identify which part of the adaptation algorithm is responsible. Layered evaluation has been proposed to address this problem (Brusilovsky *et al.* 2001). In the layered evaluation approach, adaptation is split into two phases, namely interaction assessment and adaptation decision making. The interaction assessment phase involves assessing the user-computer interaction in order to determine whether adaptation is necessary (e.g. the user is unable to complete a task). In the adaptation decision making phase, specific adaptations are selected to improve some aspect of the user's interaction with the system. Weibelzahl and Weber (2002) extend this approach further, identifying four layers to be evaluated, namely the reliability and validity of input data, inferences made by the system, adaptation decision making and the user's interaction with the system (does adaptation result in user confusion or dissatisfaction). The middle two layers are roughly equivalent to those identified by Brusilovsky *et al.* (2001).

Several qualitative and quantitative methods have been used to evaluate AUIs. Van Velsen *et. al'*s (2006) survey of AUI evaluations revealed that questionnaires (73%), interviews and interaction logging (both 42%) are the most popular methods. Several evaluation methods are usually employed in combination with one another. Van Velsen *et al.* (2007) compared think-aloud, questionnaires and interviews as methods for evaluating AUIs. They advocated the use of these methods in combination with one another, mentioning that questionnaires provide useful feedback on usability, without the positive bias associated with interviews (they found that users were less likely to provide negative feedback regarding adaptation in interviews).

Jameson (2003) stresses the importance of not just evaluating the accuracy of adaptations performed, but also the usability of these adaptations and especially whether the problems and shortcomings associated with AUIs (Section 3.5) are evident. He identified four broad techniques for evaluating AUIs, namely:

- *Wizard of Oz:* Where a human operator manually adapts the system, giving the impression that the system is intelligently adapting to the user's behaviour. This can be useful for simulating adaptations which could take a long time to implement.
- *Simulations using data from a non-adaptive system:* Where previously observed behaviour data from a non-adaptive version of a system can be used to train an adaptive version.
- *Controlled studies:* Where multiple versions of a system can be compared.
- *Studies of actual system use:* Where users can use the system over a longer time period. Observation, interviews, questionnaires and log files can be used to evaluate the system.

Controlled studies were identified as useful for uncovering usability issues and comparing different system variants. Studies of actual system use were identified as being important tools for assessing the overall usefulness and usability of AUIs and for taking into account contextual factors which influence the everyday use of such systems. Chin (2001) advocates the need for pilot studies in any evaluation of adaptive systems in order to ensure that resource intensive evaluations are conducted correctly.

A wide variety of metrics have been used to evaluate AUIs. These metrics depend to a large degree on the type of AUI being evaluated (Section 3.9). The most popular variables measured in the evaluation of AUIS are usability (53%), perceived usefulness (41%) and user behaviour (39%) (van Velsen *et al.* 2006).

Langley (1999) identifies the following four general classes of dependent variables to be considered in the evaluation of AUIs:

- *Efficiency:* AUIs are often concerned with helping users complete tasks more quickly, or using less effort than they would require if they were to complete the same task on their own. Efficiency can therefore be measured using the time taken to complete the task in question or the number of user actions required.
- *Quality:* Recommender and information filtering systems (Section 3.9.3) can be evaluated according to the quality of the recommendations they provide to the user. Quality can be difficult to measure, as it is dependent on the user's criteria for determining a quality solution (e.g. the cheapest book, or the latest stock price).
- *User Satisfaction*: User satisfaction questionnaires can be used to directly elicit feedback regarding user satisfaction with the system or the adaptations it provides. The Questionnaire for User Interface Satisfaction (QUIS) (Chin *et al.* 1988) has previously been used to measure user satisfaction with AUIs (Strachan *et al.* 2000).
- *Predictive Accuracy:* Accuracy can be used to evaluate system recommendations (Section 3.9.2). This is sometimes used as a substitute measure for quality, but should not be used alone for this purpose.

Several metrics have also been developed to specifically evaluate recommender systems. These evaluation metrics are of particular relevance to this research, as the information adaptation implemented in MediaMaps makes use of recommendations regarding the sorting of media items. These metrics include (Gena 2005):

- *Precision and Recall:* These two metrics can be used to evaluate the recommendations provided and are based on true positives (TPs, the number of system recommendations the user has accepted), false positives (FPs, the number of system recommendations the user has rejected, true negatives (TNs, recommendations that are not relevant to the user and are not suggested) and false negatives (FNs, recommendations that are relevant to the user but have not been suggested).
    - *Precision* is measured as the ratio between the relevant recommendations provided to the user and the total recommendations provided.

$$Precision = \frac{TP}{TP+FP}$$

- *Recall* is measured as the ratio between the relevant recommendations presented to the user and the total number of recommendations which should be presented to the user.

$$Recall = \frac{TP}{TP+FN}$$

- *Error rate (ε):* This can be calculated as the ratio between incorrect system behaviour (false negatives and false positives) and all system behaviour (including false positives and negatives and true positives and negatives).

$$\varepsilon = \frac{FN+FP}{TP+FP+TN+FN}$$

- *Coverage:* The percentage of items for which the system can provide recommendations. This is more relevant for content recommendations. Some items may never be recommended because of the nature of the algorithm in question.

Precision and recall have both been used to evaluate algorithms for determining the boundaries between photo collections (Naaman *et al.* 2004). In this context, precision is interpreted as:

$$Precision = \frac{correctly\ \ detected\ \ boundaries}{total\ \ number\ \ of\ detected\ \ boundaries}$$

Recall can be expressed as:

$$Recall = \frac{correctly\ \ detected\ \ boundaries}{total\ \ number\ \ of\ \ ground\ \ truth\ boundaries}$$

Another metric, which combines the precision and recall in a single metric, called the F-score (Cooper *et al.* 2005) is sometimes used.

The F-score is expressed as follows:

$$F\text{-}Score = \frac{2\ x\ precision\ \ x\ recall}{precision\ +recall}$$

6.2.3 Summary

The above discussion has identified several important trends, including:

- The importance of field studies in the evaluation of both mobile systems and AUIs;
- The importance of pilot studies as a precursor to evaluations, both in the fields of mobile HCI and AUIs, in order to avoid unnecessary effort and expense being incurred in lengthy and potentially complex evaluations which are flawed;
- The difficulties in attributing benefits achieved (or not achieved) to adaptations implemented and possible techniques for overcoming these difficulties;
- Usability, usefulness and user behaviour are the three metrics of AUIs which are evaluated most often;
- The importance and widespread use of interviews, questionnaires and interaction logging in the evaluation of AUIs;
- The importance of not just evaluating the accuracy of adaptations, but also the usability of the adaptations and the AUI system in general; and
- The use of appropriate metrics given the types of adaptation which are performed.

6.3 Evaluation Goals

MediaMaps was developed as a proof-of-concept of the Proteus Model, facilitating information, interface and visualisation adaptation. These three forms of adaptation therefore need to be the central focus of any evaluation. In particular the following was evaluated:

- The *usability* of MediaMaps, specifically whether this adaptive MMV system provides an effective means of facilitating user tasks with a high level of satisfaction; and
- The perceived *usefulness* of the three forms of adaptation. Appropriate metrics need to be used to evaluate each form of adaptation individually.

Higher-level objectives of the evaluation of MediaMaps were to determine whether the model-based design of adaptive MMV systems is feasible and whether MediaMaps addresses any of the problems of mobile visualisation (Section 2.4.2) and MMV (Section 2.5.3).

6.4 Selection of Evaluation Methods

The lessons learned from existing literature regarding the evaluation of mobile applications and AUIs were considered when selecting appropriate methods for the evaluation of MediaMaps. It was clear both from related work and from the nature of the adaptations incorporated in MediaMaps that a field study of MediaMaps would be necessary in order to evaluate the usefulness of the adaptations supported. One usage session in a controlled environment would not allow the system to learn the user's preferences and hence any attempt to evaluate the adaptations would be meaningless. It was also decided to conduct a preliminary usability evaluation of MediaMaps prior to the field study, in order to evaluate the usability of the prototype and determine whether it is effective at facilitating the user's tasks. This was to ensure that no significant usability problems were encountered by field study participants.

Since the adaptations in MediaMaps were built into the system, no non-adaptive version of the prototype was available. While the interface and visualisation adaptations can be easily disabled, the information adaptation regarding the sorting of media items into collections is a fundamental aspect of the system and disabling this feature would mean that the two systems being evaluated would have significant differences, making it difficult to draw meaningful comparisons between them. Layered evaluation was also considered unnecessary, as it is difficult to see how the interaction assessment layer could be evaluated independently. Unlike the class of systems described by Brusilovsky (2001), adaptation in MediaMaps is not based on the identification of user problems or difficulties, but rather on the identification of user tasks, which can be done with no risk of error. Furthermore, no significant assumptions are made based on user behaviour, but rather preferences are calculated based directly on user interaction data. As a result, such an evaluation was considered unnecessary.

Evaluation of the adaptations and not just the usability of the prototype was considered important. In order to combat the fact that a comparative evaluation was not possible, it was decided to directly evaluate all three forms of adaptation through a combination of interaction logging and user satisfaction questionnaires. Interaction logging is useful in order to analyse the accuracy of the information and interface adaptations in MediaMaps, while the use of user satisfaction questionnaires allows the perceived usefulness and usability of all three forms of adaptation to be evaluated. Although a direct link between the usability of the system and the adaptations performed will not be able to be drawn, the accuracy and perceived usefulness of the adaptations implemented can be directly evaluated (accuracy is only applicable to information and interface adaptation).

The preliminary evaluation of MediaMaps will be discussed first, followed by the field study.

6.5 Preliminary User Testing

Prior to the main field study of MediaMaps, preliminary user testing was conducted. The objectives, test procedure and results are now discussed.

6.5.1 Objectives

Preliminary user testing was conducted in order to evaluate the usability of MediaMaps. This was done in order to ensure that any usability problems which could have prevented the successful completion of the field study were addressed prior to its commencement.

6.5.2 Metrics

Usability can be defined as (ISO9241-11 1998):

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context."*

Usability is thus defined as consisting of three elements, namely effectiveness, efficiency and satisfaction, which can be defined as follows (NIST 2001):

- *Effectiveness: "The accuracy and completeness with which users achieve specified goals."*
- *Efficiency: "The resources expended in relation to the accuracy and completeness with which users achieve goals."*
- *Satisfaction: "Freedom from discomfort, and positive attitudes towards the use of the system."*

Effectiveness and user satisfaction were measured in this evaluation. Efficiency was not considered, as this would require tasks to be performed repeatedly to produce meaningful results (Tullis and Albert 2008). In addition, it was deemed important not to put unnecessary time pressure on participants during the evaluation, but rather to encourage them to explore and use MediaMaps at their own pace in order to assess the effectiveness and satisfaction of the system. Information adaptation (sorting of media items into collections) was considered in this evaluation, but a longitudinal evaluation of the system was required to evaluate the visualisation and interface adaptations (Section 6.6).

6.5.3 Instruments

Effectiveness was measured through the use of questions which were included in the test plan to determine whether participants were able to successfully and accurately complete the required tasks (Appendix A). Participants were also observed to see if they experienced any difficulties in completing the assigned tasks and to determine whether they understood the information adaptation provided by the system (in terms of recommendations regarding whether a new media item belongs in a new or existing collection). User satisfaction was measured through the use of a user satisfaction questionnaire that participants were asked to complete at the end of the evaluation. This questionnaire (Appendix B) was based on the Questionnaire for User-Interface Satisfaction (QUIS) (Chin *et al.* 1988). A five point Likert scale with antonyms at either end of the scale was used for most questions, with open-ended questions at the end to obtain positive, negative and general feedback.

The post-test questionnaire was modified from the original QUIS and was divided into the following five sections:

- Interface Design;
- Navigation and Browsing;
- Searching for Media;
- Capturing Media; and
- Sorting Media into Collections;

A logging mechanism was also built into the system to record user interaction. A Nokia N95 with MediaMaps pre-installed was used for testing purposes.

6.5.4 Selection of Participants

Participants were selected to reflect a cross section of the typical user profile of experienced smart phone users (Barnum 2002). A total of ten participants (six male and four female) took part in the evaluation. All of the participants were experienced mobile phone users, most using mobile phones on a daily basis for more than five years. Most of the participants had at least two years experience with camera phones. Although most participants used their mobile phones to record multimedia infrequently (about once a month), all the participants had previous experience in using mobile phones to record multimedia. The participants' ages ranged from 22 to 50 (mean = 31) and they are employed in a variety of occupations. Figure 6.1 shows more detail regarding the participants' camera phone experience and occupations. None of the test participants owned a Nokia N95 (or another Nokia N-series phone).

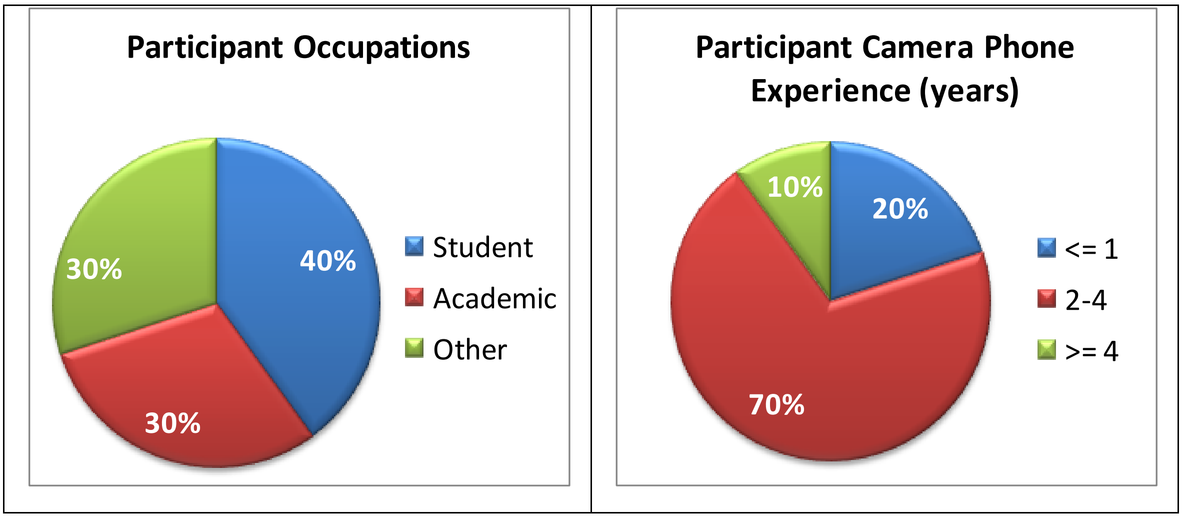

Figure 6.1: Preliminary study participant demographics (*n*=10)

6.5.5 Tasks

Participants were provided with a version of MediaMaps with several existing media collections already created using the system and available for browsing. The test plan (Appendix A) required them to complete the following basic tasks:

- Browse an existing media collection in the static map view;
- Browse an existing media collection in the animated map view;
- Customise the map visualisation in terms of the map type used and the media types visualised;
- Search for a media collection created on a particular date;
- Search for a media collection created at a particular location;
- Capture at least five photos and two videos;
- Locate and view the first photo captured; and
- Locate and view the first video captured.

6.5.6 Test Procedure

User testing was conducted at a variety of locations in Port Elizabeth, both indoors and outdoors (and in some cases, both). Participants were encouraged to move around while completing the task list in order to capture photos and videos of their surroundings. Participants were observed by the test administrator while using MediaMaps and user comments were recorded. Upon completion of the test, participants were required to complete a user satisfaction questionnaire (Appendix B). A more detailed version of the test procedure is provided in Appendix C.

Participants were free to complete the test plan at their own pace and discussion and comment was encouraged throughout the evaluation. The administrator only provided assistance where it was obvious that the participant was unable to successfully complete a task. This was noted in order to ensure that the effectiveness data reflected this assistance.

6.5.7 Results

The results of the evaluation were categorised into performance metrics (effectiveness) and satisfaction metrics (user satisfaction).

6.5.7.1 Performance Results

The answers obtained to the questions embedded in the test plan showed the system to be effective in supporting participants in completing their tasks (Table 6.1). All but two of the responses were correct, with the incorrect responses relating to the incorrect identification of the name of a media collection, as a result of this not being displayed prominently enough. This corresponds to an accuracy of over 97% in the responses received. All major system tasks, including capturing media and browsing media collections, were completed easily.

The high level of accuracy of participant responses to the questions embedded in the test plan, as well as the observation of the participants using the system suggest that MediaMaps was highly effective in supporting the participants' tasks. Participants experienced few problems capturing, browsing and searching for media. Participants were also quick to understand and become accustomed to the information adaptation facilitated by the system.

Task	Accuracy (%)
1. Identifying the number of media items in a collection	100
2. Identifying the number of media items in a collection (post-customisation)	100
3. Identifying the name of a collection (searching by date)	90
4. Identifying the number of photos in a collection (searching by date)	100
5. Identifying the name of a collection (searching by location)	100
6. Identifying most recently captured item (searching by location)	100
7. Identifying the name of the first participant-created collection	90
8. Identifying the name of the first participant-captured photo	100
9. Identifying the name of the first participant-captured video	100
Overall	97.78

Table 6.1: Accuracy of participant responses to questions embedded in the task list (*n*=10)

A few minor usability problems were identified during the course of the evaluation. Some participants did not notice the use of red and green to denote older and newer items on the map (Figure 5.6) until this was pointed out to them. Some participants also experienced difficulty in selecting collections or items when these were location-tagged at the same location. When this occurred, several icons were drawn directly on top of each other, resulting in the occlusion of the icons underneath.

6.5.7.2 Satisfaction Results

A complete summary of the feedback from the user satisfaction questionnaire is included in Appendix D. Figure 6.2 shows the results from the post-test questionnaire summarised according to the five sections of the questionnaire (Section 6.5.3). Ninety-five percent confidence intervals are shown for the mean values.

High levels of user satisfaction were reported by the participants in all sections of the questionnaire. Mean and median ratings for all sections were at least 4. Considering the objectives of this research, the very high mean ratings in Section D (Searching for Media), Section E (Capturing Media) and Section F (Sorting Media) are of particular interest. These sections correspond to major system tasks related either directly or indirectly to the information adaptation performed by the system. User satisfaction with all three of these sections was very high, which provides evidence to support the benefits of using information adaptation in MMV systems.

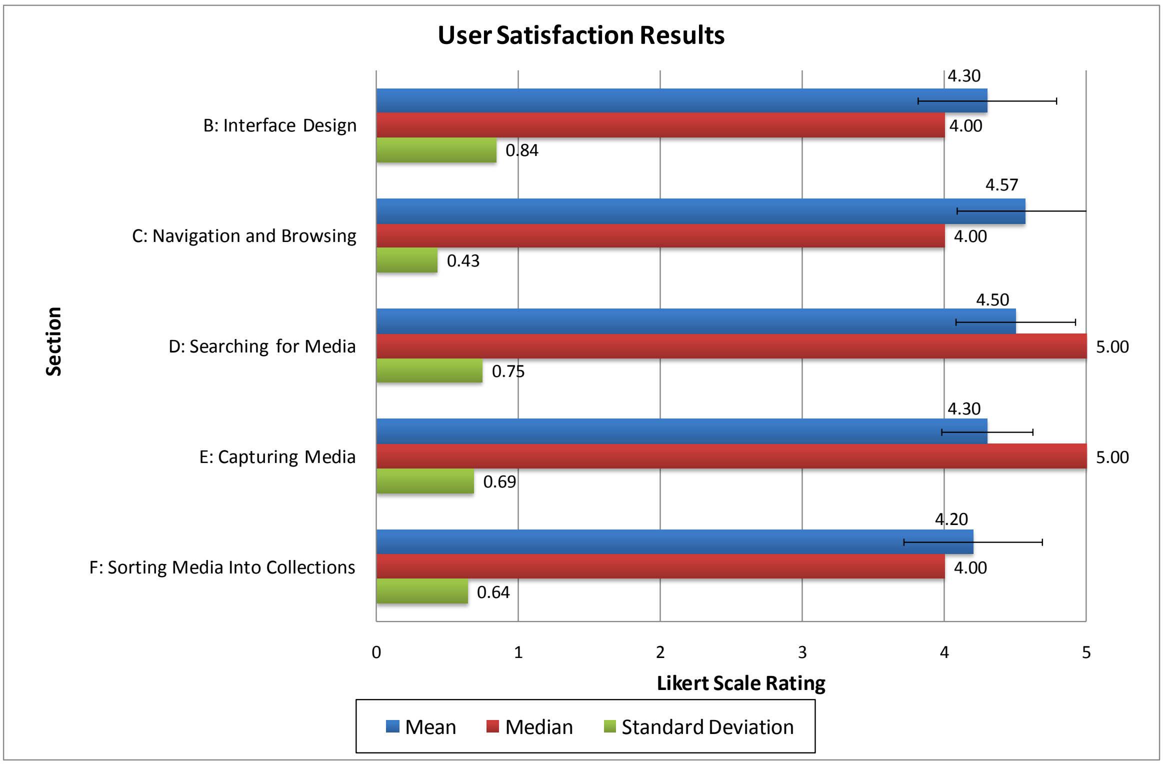

Figure 6.2: Summary of post-test user satisfaction questionnaire results by section (*n*=10)

Section F (Sorting Media into Collections) related specifically to the information adaptation facilitated by the system. The mean, median and standard deviation values for this section are shown in Figure 6.3. Ninety-five percent confidence intervals are shown for the mean values.

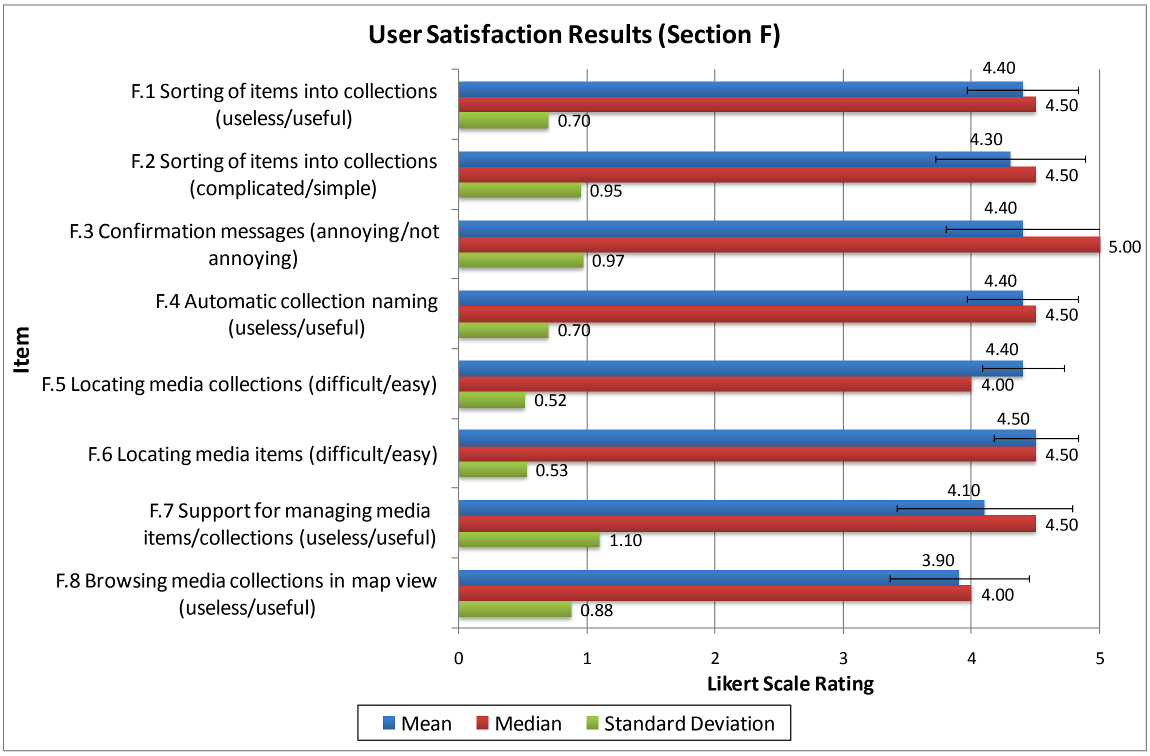

Figure 6.3: User satisfaction feedback for Section F (Sorting Media Items into Collections) (*n*=10)

Nine out of the ten participants rated the adaptive sorting of items into collections as at least a 4.00 (on the five point Likert scale) in terms of usefulness (F.1 in Figure 6.3). The simplicity of the semi-automatic sorting process (F.2) was also rated favourably with a mean value of 4.30 (with 1=Complicated and 5=Simple). The high mean rating of 4.50 for locating individual media items suggests that participants found it very easy to locate media that was sorted into collections (F.6). Only browsing of media collections in the map view (F.8) had a mean value of below 4.00. General comments and observations suggest that this was due to the amount of scrolling required and other minor usability problems.

General feedback was also very positive. Negative comments were restricted to minor usability issues mostly related to the user interface. The only negative feedback regarding information adaptation was from one participant who suggested that more flexibility would be useful regarding the sorting of items into collections. Instead of only having a choice between the current collection and a new collection, the participant in question wanted a choice of several possible collections and the ability to later move an item to a different collection.

The participants suggested several improvements, including:

- Improved feedback when browsing the map view (a tool-tip type effect was suggested for when the cursor passes over a collection or media item on the map).
- Improved labelling for the legend indicating the use of a red/green colour spectrum to visualise the age of the media items/collections visualised (dates were initially used to indicate the start and end of the period currently being visualised, but some participants were unsure what these dates meant).
- A means of addressing the occlusion problem when multiple media items were tagged at the same, or very similar locations.
- The need for a "View on Map" option when viewing an individual media item (this was initially only available when browsing the list of collections).
- A faster default scrolling speed when browsing the map view.

Positive feedback covered a wide range of issues, with the usefulness, aesthetic design, ease of use and learnability of MediaMaps emerging as common themes. Several participants also expressed the desire to have MediaMaps installed on their personal mobile phones.

Participants also provided the following positive feedback regarding the system's semi-automatic sorting of media items into collections:

- *"Suggestion to add to existing collection very useful"*
- *"Automatic groupings make it quicker to use and file media"*
- *"Automatic tagging using GPS streamlines the capturing and sorting process"*

It was also encouraging to note that, although the ordering of list options (interface adaptation) was not specifically part of this evaluation, several participants responded positively when they returned to a long list of options and found that their recent selections had migrated to the top of the list.

The positive results of this evaluation provide evidence to suggest that AUIs could provide an effective and satisfying means of improving MMV systems. Further, longitudinal evaluation of the system was needed to support these findings and to investigate the accuracy of the information and interface adaptations, as well as the user satisfaction with all three types of adaptation implemented in MediaMaps. A field study was conducted to determine this, the results of which are outlined in Section 6.6. Several minor usability problems and opportunities for improvement were identified during the preliminary evaluation. These were addressed by making several modifications to MediaMaps. These modifications are discussed in the following section.

6.5.8 Prototype Modifications

The goal of the preliminary evaluation was to identify usability problems prior to conducting a field study of the system (Section 6.5.1). In response to the qualitative feedback received during the preliminary evaluation of MediaMaps, several modifications were made to the prototype in order to address the problems that were encountered. These modifications were:

- A tool-tip type effect was added in order to provide improved feedback when browsing the map view. When the cursor moves over a media item, the item's name and date of capture are displayed at the top of the map view. When viewing collections, the collection name is displayed. This helps to prevent the situation observed during the preliminary user testing, where participants relied on trial and error to identify a media item or collection.

- The labelling of the legend indicating the use of a red/green colour spectrum to visualise the age of media items/collections was simplified. Dates were replaced with the terms “Older” and “Newer”, with dates only displayed when filtering the time period being visualised.
- The occlusion problem was addressed by detecting when icons are going to be drawn at the same x and y coordinates. These icons are then diagonally arranged centred on the x and y coordinates in question. This provides for easier selection.
- A “View on Map” option was added when viewing an individual media item in order to view the item visualised in the map view.
- An acceleration mechanism was built into the system when browsing the map view. The longer the user scrolls in a certain direction, the faster the map will scroll (up to a maximum speed). This was implemented to make scrolling long distances across the map less tedious. Unlike the Speed-dependent Automatic Zooming (SDAZ) technique (Section 2.6.2), the zoom level is not adjusted, but rather the speed at which the viewing window moves is increased.

Several other minor issues were corrected, including preventing users from accidentally exiting the system and replacing the block cursor (used when manually selecting a location) with a crosshair cursor.

6.6 Field Testing

After completing the changes mentioned above, the updated version of MediaMaps was pilot tested to ensure that no new bugs or usability problems had been introduced into the system. Positive feedback was received regarding the changes made to the system. After a few weeks of testing to ensure stability, a field study of MediaMaps was conducted.

6.6.1 Objectives

The field study was conducted to evaluate the adaptations incorporated in MediaMaps over a longer time period than was done in the previous evaluation. All three forms of adaptation are such that the resulting effects are only noticeable after using the system on multiple occasions. As with previous AUI research, the usability and perceived usefulness of the information, visualisation and interface adaptations in MediaMaps were considered to be of particular interest (Section 6.3).

6.6.2 Metrics

Several metrics were used to evaluate the three forms of adaptation. These were:

- *Accuracy:* Information and interface adaptation were evaluated in terms of the accuracy of the performed adaptations in order to determine whether these matched the user's needs. The *Precision, Recall* and *F-score* metrics (Section 6.2.2) have all previously been used to evaluate the accuracy of algorithms for detecting boundaries between photo collections and were therefore considered appropriate metrics for evaluating the information adaptation in MediaMaps. Accuracy has previously been identified as a key need for effectiveness and is also important in order to achieve efficiency (Stone *et al.* 2005).

- *Learnability:* MediaMaps was evaluated in order to determine whether users became more proficient over time. A common way of measuring this is to examine task completion times as a function of experience (Tullis and Albert 2008). Learnability is often identified as a dimension of usability, in addition to those discussed in Section 6.5.2 (Nielsen 1994b; Quesenbery 2003)

- *User Satisfaction:* Several aspects of user satisfaction were measured, regarding the system in general and the adaptations in particular.

6.6.3 Instruments

The participants were provided with a high-level test plan (Appendix E), describing the system, the adaptations facilitated and user tasks. User interaction was logged in order to later evaluate the accuracy of the information and interface adaptations. Task completion times were also calculated, in order to evaluate learnability. A post-test user satisfaction questionnaire (Appendix F), based on the QUIS was used to elicit user satisfaction feedback. The questionnaire used for the field study differed significantly from the one used for the pilot study, with less focus on general usability and more focus on the three forms of adaptation facilitated.

The field study questionnaire was broken down into the following sections:

- Capturing media;
- Searching for media;

- Sorting media into collections (information adaptation);
- List sorting (interface adaptation); and
- Map view adaptations (visualisation adaptation).

6.6.4 Selection of Participants

International test participants were recruited using a bidding process (Mob4Hire 2008). Twenty-six bids were accepted, although only 20 test participants (14 male, 6 female) completed the field study. This gender split is roughly equivalent to the split in the smart phone sector of the market (Comscore 2008). Non-completion occurred for a variety of reasons, including technical problems with the participants' phones and insufficient use of the system to warrant useful feedback. An average of $46.50 (US) was paid per test participant.

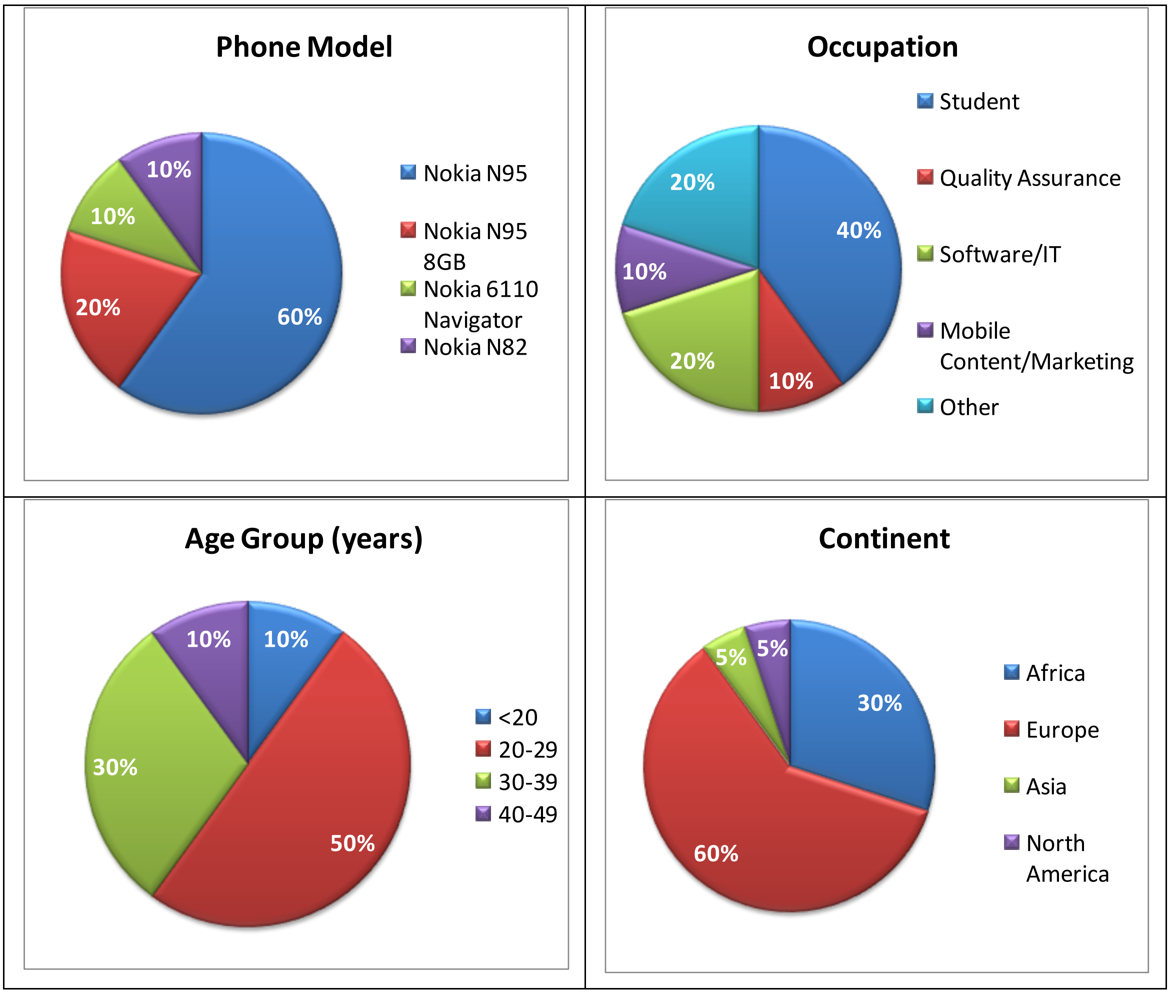

Figure 6.4: Demographic profile of field study participants (*n*=20)

Participants were recruited from twelve different countries on four continents. Figure 6.4 shows the demographic profile of the test participants. The participants all had at least three years of mobile phone experience and all but one participant had at least three years of experience using a camera phone. All participants rated themselves as either intermediate or expert phone users. The group of test participants therefore matches the target user profile of experienced smart phone users. Nokia N-series phones equipped with GPS receivers were targeted as these provide the necessary hardware and support for the required Java ME APIs.

6.6.5 Tasks

The field study test plan was much less restrictive than the test plan for the preliminary user testing. Participants were free to use the system as desired, as long as the following tasks were performed throughout the three week test period:

- *Capture media:* Participants were required to use MediaMaps as their primary media capturing software for the duration of the test period, capturing multiple photos, videos and sound recordings. Participants were encouraged to capture at least 40 media items. This was not enforced as a rule, but rather served as a guideline to ensure minimum requirements were met during the test period.
- *Browse media collections:* Media collections and individual media items captured using MediaMaps were required to be browsed in both the map view and the list view.
- *Search for media:* Participants were required to make use of the search functionality implemented in MediaMaps to search for media items by date and by location name.
- *Customise settings:* Participants were encouraged to make use of the options available to customise the map-based visualisations, including customising the map type, data types and time period.

6.6.6 Test Procedure

Participants were required to use MediaMaps on their personal mobile phones for a period of three weeks. The fact that participants used the system on their personal phones ensured that any usability issues encountered were not because participants were unfamiliar with the mobile phone in question. A text message was sent to the participants’ phones containing a link to download the installer files for the application. Participants were provided with the test plan and the post-test satisfaction questionnaire at the beginning of the field study in order to ensure that they knew what was expected of them. Support was provided to ensure that all participants were able to install the prototype successfully and get it up and running.

MediaMaps, as a signed Java MIDlet, requires various permissions to be set on the user's phone in order to allow the system to be used without constant warning messages. Some participants required assistance completing this important step (not setting the required permissions could have a serious negative impact on usability).

Contact was maintained with test participants during the test period through the use of email, instant messaging, phone contact and (in the case of local participants) face to face contact in order to provide support where necessary. Upon completion of the three week test period, participants were required to complete the post-test satisfaction questionnaire and submit it, along with the interaction log files stored on their phone. Additional feedback beyond that required in the questionnaire was encouraged and in some cases informal discussions were conducted with test participants upon completion of the field study regarding their experiences with MediaMaps.

6.6.7 Results

The discussion and analysis of the field study results is divided into performance results and satisfaction results.

6.6.7.1 Performance Results

Two performance metrics were used in the field study of MediaMaps. The first of these addressed the learnability of the system. The uncontrolled nature of the field study meant that the logging mechanism embedded in the system had to be used to record the time when different actions were completed. Task times for what was considered the most complex system task (namely the capture, location-tagging and sorting of media items into collections) were used to measure learnability. This task was also selected because it involves information adaptation. Most other system tasks were either too simple, or did not have clear enough start and end events to make measuring completion times meaningful. The uncontrolled nature of the field study means that the large degree of variation in the times is not surprising. In order to remove extreme cases (where the participant was clearly distracted at some point in completing the task), task times of greater than or equal to a minute were removed from the data before it was analysed. Cases where the process was cancelled before completion (e.g. a participant rejected a photo without tagging it) were not considered.

Aggregated and averaged task completion times for all participants are shown in Figure 6.5 with a linear trend line fitted. A clear negative trend is evident, showing the average completion time decreasing from around the 40 second mark to 20 seconds to complete the capture, tagging and sorting of a media item. A significant correlation co-efficient of -0.77 (p-value < 0.05) was calculated between the completion time for this task and the number of media items captured, indicating a clear negative trend in the task times for this task as the number of media items captured increased. This reflects favourably upon the learnability of the system and indicates that participants became accustomed to the process involved in capturing and tagging photos, as well as dealing with system recommendations regarding the sorting of these items, over time.

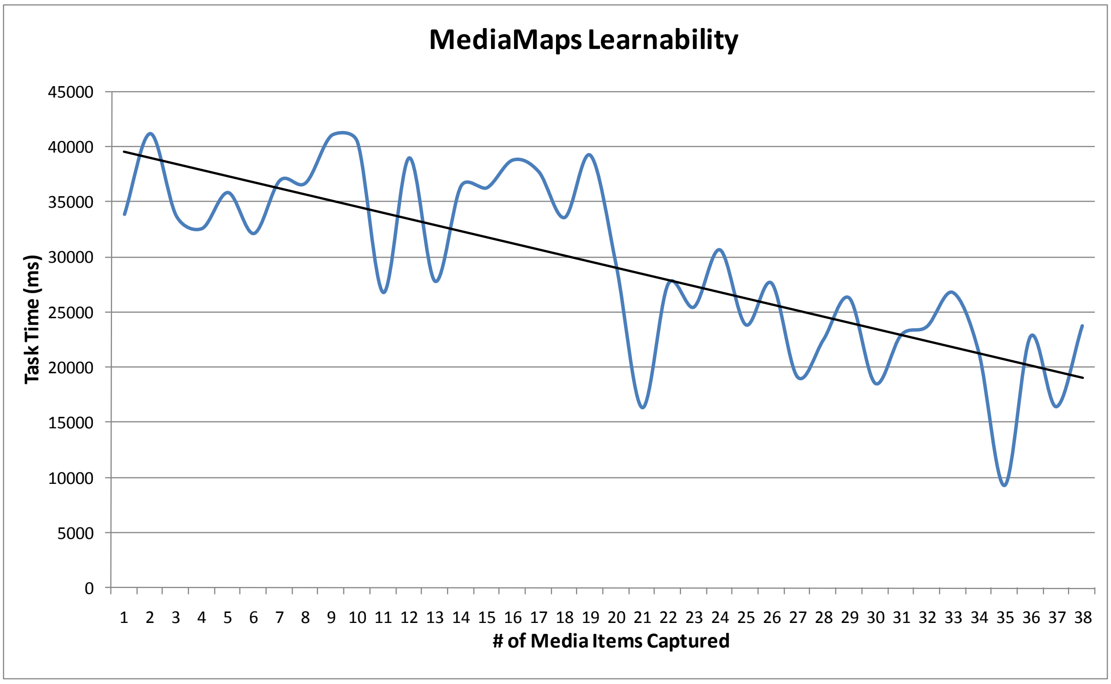

Figure 6.5: Task completion time vs. number of media items captured using MediaMaps

Accuracy was the second performance metric evaluated. The accuracy of both the information adaptation (sorting media items into collections) and interface adaptation (ordering of list options) was analysed. The precision, recall and F-score metrics (Section 6.2.2) were used to assess the accuracy of the RED algorithm used to perform information adaptation and were calculated using the log files obtained from the test participants.

The precision of the algorithm was calculated as the ratio between the correctly detected boundaries (when MediaMaps recommends that the item belongs to a new collection and the user agrees) and the total number of detected boundaries (both correct and incorrect recommendations that an item belongs in a new collection). Recall was calculated as the ratio between the correctly detected boundaries and the actual number of boundaries (the combination of correctly detected boundaries and unidentified boundaries). Table 6.2 summarises the accuracy of the RED algorithm used to implement information adaptation in MediaMaps. The cumulative values were calculated across all participants' data (in which case participants who captured more media items influence the value to a greater degree), while the average values were calculated by taking the mean of the values calculated for individual participants (all participants therefore have equal weighting).

	Cumulative	Average
Accuracy	89.34%	87.44%
Precision	81.52%	84.81%
Recall	76.53%	81.14%
F-Score	78.95%	83.33%

Table 6.2: Accuracy results for the RED algorithm implemented in MediaMaps

General accuracy was measured as the percentage of system recommendations accepted by the participants. Overall accuracy was higher than all three other metrics shown in Table 6.2. This was to be expected, as both the precision and recall metrics focus specifically on where the algorithm chooses to place the boundaries between different events. In some instances, this will be an easy decision, as there is a large time gap between successive media items being captured. In other cases the decision is not as clear. The user's perception of what belongs in a new event or new collection may be unpredictable and difficult for the algorithm to learn, resulting in lower precision and recall statistics. The fact that the recall values are lower than the precision values indicates that errors made by the RED algorithm were more often as a result of false negatives (a new event was not detected when one should have been) than false positives (a new event was detected when one should not have been).

Evaluations of other algorithms for performing clustering photo collections based on time (and in some instances locations) have revealed maximum precision and recall values in the mid-80 percent range and maximum F-Scores (which combine both precision and recall) of about 85% (Naaman *et al.* 2004; Cooper *et al.* 2005). Cooper *et al.* (2005) report a maximum F-Score of 85.68% using their temporal-based similarity clustering algorithm. It should, however, be noted that these algorithms process photo collections after these are captured and can therefore have the advantage of more knowledge regarding the structure of the collection. The RED algorithm (implemented in MediaMaps) performs clustering incrementally and in real-time and can therefore only rely on the user's previous photo taking behaviour. Post-processing algorithms can look at the characteristics of the entire photo collection, looking both backwards and forwards from a particular photo in order to determine where to place the boundaries between events. Furthermore, the algorithms mentioned above were tested on only two or three photo collections. The RED algorithm in MediaMaps was tested on the media collections of 20 different users, with large variations in user behaviour. The average F-Score of 83.33% which was obtained therefore represents a significant achievement, given the incremental, real-time nature of the RED algorithm, as well as its low-resource requirements. These results suggest that the adaptive nature of the RED algorithm helped keep accuracy levels high despite the varying preferences and behaviour of different users.

The accuracy of the RED algorithm was also analysed according to the three regions (Section 5.8.2) into which newly captured media items are classified. Table 6.3 shows a summary of the accuracy percentages achieved for the individual regions of the algorithm. Only accuracy is shown, as it is not possible to calculate meaningful precision and recall statistics for the individual regions.

	Cumulative	**Average**
R_o (New media item identified as belonging to the current collection)	81.82%	87.50%
R_l (Intermediate zone, location information incorporated)	91.87%	85.55%
R_n (New media item identified as belonging to a new collection)	80.77%	89.96%

Table 6.3: Accuracy of the RED algorithm for different classification regions

Very good accuracy levels (both cumulative and average accuracy) were achieved across all three regions. Of particular interest was the high level of accuracy achieved in the intermediate region (R_l), where the distance between the newly captured media item and the previous media item is used to determine whether the current item belongs in a new collection, or in the current collection. Decisions regarding partitioning of media items that fall into this region are the most difficult, as it is not clear based on the time gaps whether the current media item belongs to a new collection or not. It should be noted that only 7% of media items captured fell into this region. Nevertheless, the high accuracy levels achieved indicate that incorporation of location information assists the algorithm in making accurate decisions in this intermediate region.

The RED algorithm was incorporated into MediaMaps in order to provide information adaptation by adapting to user behaviour. According to our AUI definition (Section 3.2), the algorithm would be expected to improve in response to user feedback. Figure 6.6 shows an aggregated plot of the accuracy of the RED algorithm versus the number of media items captured, for all field study participants. A logarithmic trend line was fitted, showing that the output does exhibit accuracy improvements over time. This trend was clearly evident in the data from some participants, while in others, the accuracy varied dramatically. As a result, the graph in Figure 6.6 exhibits a large degree of variance. In general, however, learning was evident, with gradual improvements in accuracy following user corrections.

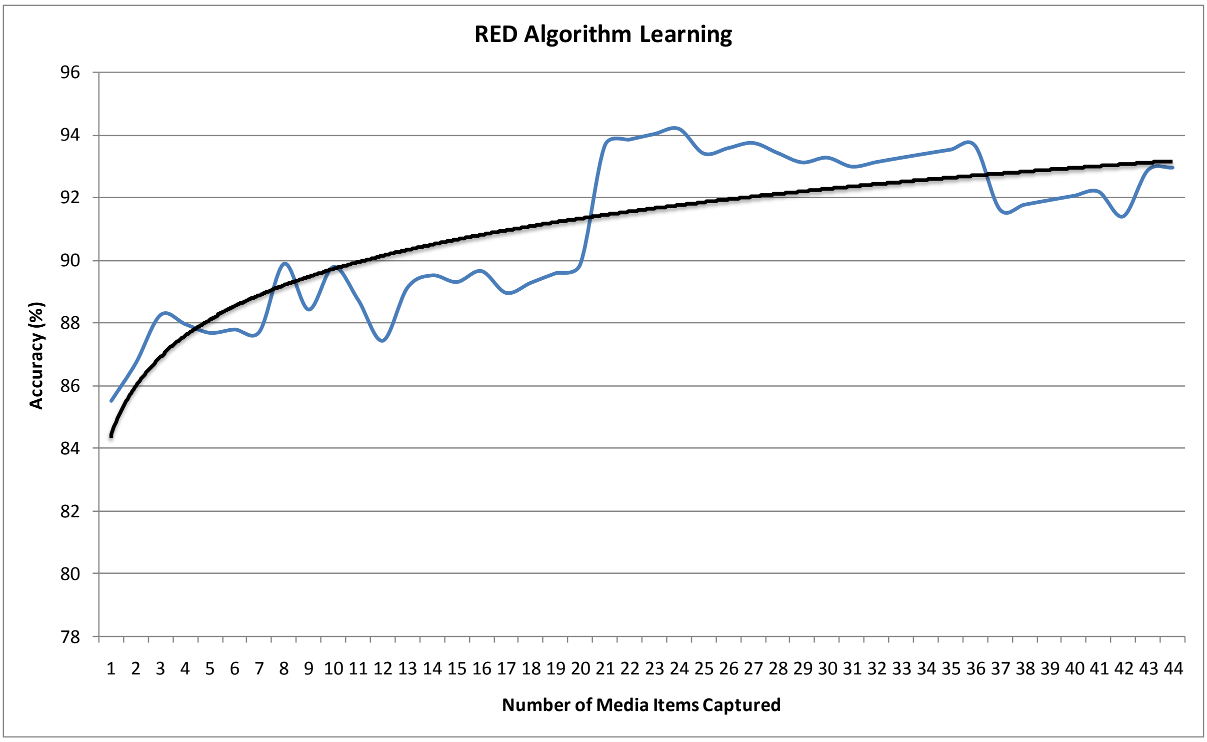

Figure 6.6: Recommendation accuracy vs. number of media items captured

For all participants, a significant correlation co-efficient of 0.79 (p-value < 0.05) was calculated between the number of media items captured and the accuracy percentage achieved. This indicates a strong positive correlation between the number of media items captured and the accuracy of the recommendations provided, suggesting that the algorithm improves in response to user feedback and that accuracy improves the longer the system is used.

Accuracy was also assessed for the interface adaptation involving the ordering of list options (Section 5.8.3). It should be noted that many of these lists grow as the system is used (e.g. the list of media collections). As a result, the promotion of recently and frequently selected items to the top of the list is likely to become more useful as the lists grow. Bearing this in mind, list selections were divided into three categories:

- *Pre-Adaptive Selections:* list selection on lists of three or less items;
- *Adaptive Selections:* list selections from the adaptive top portion of the list; and
- *Non-Adaptive Selections:* list selections from the non-adaptive portion of the list.

Figure 6.7 shows list selections split into adaptive and non-adaptive selections (pre-adaptive selections were not considered). A cumulative accuracy of 77.37% was achieved for all participants and an average accuracy of 76.78%. This high level of accuracy suggests that participants made frequent use of the promoted options. Thus, by providing easy access to the options users are interested in, interface adaptation allows user list selection to be more efficient. It might be argued that this accuracy may be biased by the initially short length of some of the lists. A slight positive correlation was, however, calculated between the number of selections (suggesting more use and likely longer lists) and the accuracy of the adaptive portion of the lists.

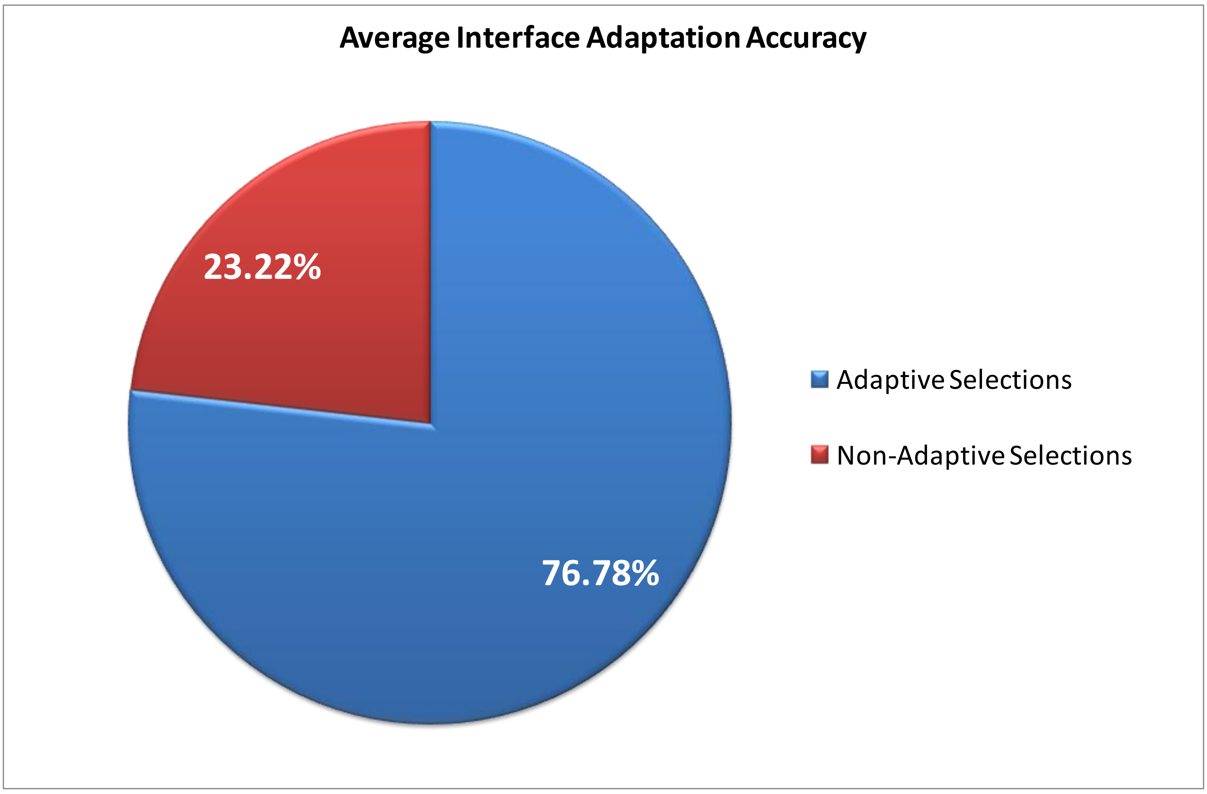

Figure 6.7: Average interface adaptation accuracy

6.6.7.2 Satisfaction Results

A complete summary of the feedback from the field study user satisfaction questionnaire is included in Appendix G. Figure 6.8 shows the feedback from the post-test questionnaire summarised according to the five sections of the questionnaire (Section 6.6.3). Ninety-five percent confidence intervals are shown for the mean values. Five-point Likert scales were used in the quantitative sections of the questionnaire, with antonyms at the extremes of the scales.

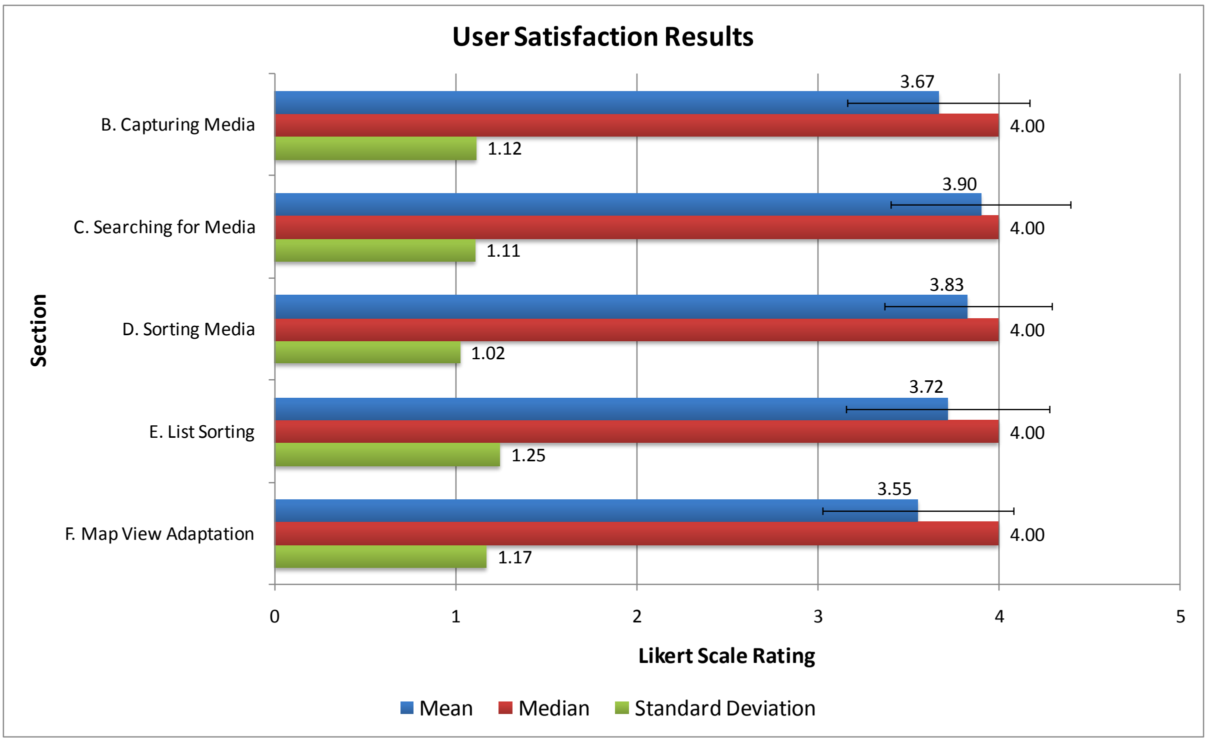

Figure 6.8: Field study user satisfaction questionnaire results summarised by section (*n*=20)

The questionnaire was divided into five quantitative feedback sections (Section A was concerned with biographical information and Section G elicited general qualitative feedback). Sections B and C related to general usability of the major system functions of capturing media and searching for located media and the ease of performing these tasks in MediaMaps. Sections D-F evaluated the three forms of adaptation. The terms "Sorting Media" (Information Adaptation), "List Sorting" (Interface Adaptation) and "Map View Adaptation" (Visualisation Adaptation) were used to refer to the three types of adaptation in the questionnaire in order to prevent any user confusion. As recommended by Jameson (Section 6.2.2), not just the accuracy (Section 6.6.7.1) and perceived usefulness of the adaptations were considered, but also the usability of these adaptations.

In Section 3.5, four major problems and shortcomings of AUIs were identified which need to be offset against the potential benefits (Section 3.4) they provide. These four problems were considered as follows in the evaluation of MediaMaps:

- *Privacy:* As MediaMaps is a single user system, with user interaction data only stored locally on the user's personal mobile device, privacy is not an issue.
- *Confusion:* A problem which often occurs in AUI systems is that users can become confused by adaptations performed autonomously by the system. Users often fail to understand why these adaptations are taking place. In order to evaluate whether this was the case, users were asked to rate all three forms of adaptation on a Likert scale (where 1=Confusing and 5=Logical). MediaMaps provides a feature to disable adaptations (Section 5.8.3).
- *Learnability:* The learnability of MediaMaps was addressed in Section 6.6.7.1. Further qualitative feedback in this regard is discussed at the end of this section.
- *Obtrusiveness:* AUIs which provide task assistance or provide recommendations also encounter problems because this system intervention may be perceived as annoying by the user. In order to evaluate this aspect of MediaMaps, users were required to rate the recommendations provided by the RED algorithm on a five-point Likert scale (where 1=Annoying and 5=Not annoying). Furthermore, in order to determine whether participants actually noticed adaptation taking place, participants were asked to rate all three forms of adaptation on a five-point Likert scale (where 1=Not noticeable and 5=Noticeable).

Most importantly, user perceptions regarding the usefulness of all three forms of adaptation were also elicited. There is little point in performing adaptation if the results are not regarded by the users as useful. After usability, perceived usefulness is the second-most evaluated aspect of AUIs (Section 6.2.2). The usefulness of information adaptation was evaluated in four respects, namely the sorting of media items into collections, automatic collection naming, the support for managing media items/collections and the ability to browse these collections in a map view. Participants were also asked to rate the difficulty/ease of sorting media items into collections and locating collections and individual media items.

User satisfaction results for Section B (capturing media) and Section C (searching for media) were very favourable. Only the ease of manual location-tagging of media (users are required to manually select their location when their position cannot be determined using GPS) received mean (3.05) and median (3) ratings below 4. Two possible explanations for this can be suggested. The first is that the maps used in MediaMaps (as downloaded from Microsoft Live Maps) vary in terms of the level of detail provided. In most cities the maps are extremely detailed, while in some locations the maps lack detail. This could lead to problems for users in low detail areas in determining their location. Another possible explanation concerns the place names that MediaMaps assigns to media items once they are location-tagged (Section 5.7.1.2). Despite over 1000 place names being incorporated into MediaMaps, the names assigned to media items were in some cases inaccurate (even though the place name assigned was the closest name stored in the system, this may have been hundreds of kilometres away). Discussions with participants revealed that this led to the incorrect impression amongst some participants that the manual location-tagging mode made it difficult to select their correct location. The large standard deviation (1.28) obtained supports these two explanations, as these problems would only have been encountered by some participants, depending on their physical location.

User satisfaction ratings regarding the sorting of media items into collections (information adaptation) are summarised in Table 6.4.

	Mean	**Median**	**Mode**	**St. Dev.**
D.1 Sorting of items into collections (Not Useful/Useful)	4.20	4.50	5.00	0.95
D.2 Sorting of items into collections (Difficult/Easy)	3.65	4.00	4.00	1.09
D.3 Confirmation messages (Annoying/Not Annoying)	3.40	3.50	4.00	1.14
D.4 Automatic collection naming (Not Useful/Useful)	3.80	4.00	5.00	1.19
D.5 Locating media collections (Difficult/Easy)	4.05	4.00	4.00	0.99
D.6 Locating media items (Difficult/Easy)	4.00	4.00	4.00	0.86
D.7 Support for managing media items/collections (Not Useful/Useful)	3.80	4.00	4.00	0.95
D.8 Browsing media collections in map view (Not Useful/Useful)	4.25	4.50	5.00	0.97
D.9 Sorting of items into collections (Not Noticeable/Noticeable)	3.55	3.00	3.00	1.09
D.10 Sorting of items into collections (Confusing/Logical)	3.60	4.00	4.00	0.99

Table 6.4: User satisfaction results for Section D (Sorting Media into Collections) (*n*=20).

Confirmation messages presented to the user (D.3) were rated more annoying than in the preliminary user testing (Section 6.5). Given the longer duration of the field study, this suggests that users find these messages to be more annoying over time. A more subtle means of integrating these into the functioning of the system needs to be found. All four aspects of MediaMaps related to usefulness were rated favourably (D.1, D.4, D.7 and D.8). Particularly encouraging is the high rating for the usefulness of sorting media items into collections (D.1) (mean=4.20, median=4.50 and mode=5.00). This suggests that users found information adaptation (implemented using the RED algorithm) to be very useful. Participants also expressed positive views regarding MediaMaps' support for the management of their media collections (D.7) (mean=3.80, median=4.00 and mode=4.00). The ability to browse media collections in a map view (D.8) was rated higher than in the preliminary user testing (mean=4.25 as opposed to mean=3.90), suggesting that this capability proved more useful in real-world situations than in the restricted short-term user testing environment. The suggested collection names which MediaMaps automatically generates (D.4) were also favourably received, although the fairly large standard deviation (1.19) and higher mode (5.00) than mean (3.80) suggested that opinions of this aspect of MediaMaps differed fairly widely (probably as a result of the variability in the accuracy of the suggested names depending on the user's location – as discussed above).

Participants were also asked to rate the ease of locating media collections (D.5) and individual media items (D.6) sorted into collections by MediaMaps. Participants expressed a clear view that these tasks were easy to perform in MediaMaps (D.5 mean=4.05 and D.6 mean = 4.00). This suggests that the sorting and map-based visualisation of media collections were effective at supporting participants in locating previously captured media.

Participants were also asked to rate the sorting of media items into collections on a five-point Likert scale (where 1=Not noticeable and 5=Noticeable). Ratings for this aspect of information adaptation were surprisingly low (mean=3.55, median=3.00, mode=3.00), given the associated ratings for the annoyance levels associated with the system recommendations. Either some participants did not understand the significance of these recommendations, or participants merely became accustomed to the process of capturing media and responding to system recommendations. It is possible that some participants expected additional sorting to take place after the capturing process. Nevertheless, unobtrusiveness is not an undesirable property of adaptation. Despite the slightly lower mean value (3.60), the median and mode values (4.00) for D.10 suggest that most participants found system recommendations regarding the sorting of media items to be logical. A positive correlation between media sorting recommendation accuracy and usefulness was calculated (*r*=0.33), but was not found to be statistically significant. Nevertheless, this correlation suggests a positive trend between the accuracy of the recommendations and participants perceptions' of usefulness.

A summary of user satisfaction results for Section E regarding interface adaptation (sorting of list options) is provided in Table 6.5. The usefulness of the ordering of list options, in which the most recent and two most frequent options are promoted to the top of the list was favourably rated (E.2). Most participants also found the algorithm for list ordering to be logical (E.3). Similar to information adaptation, the Noticeable/Not Noticeable aspect of interface adaptation (E.1) was rated slightly lower. If the accuracy of the adaptive lists is considered (Section 6.6.7.1 and Figure 6.7), the results of the field study suggest that the interface adaptation was accurate and useful. Correlations between interface adaptation accuracy and the user satisfaction questionnaire results were calculated, but proved inconclusive, with some small negative correlations. This suggests that participant opinions regarding the sorting of list options had more to do with their pre-conceived perceptions of the idea than actual performance.

	Mean	**Median**	**Mode**	**St. Dev.**
E.1 Ordering of lists of options (Not Noticeable/Noticeable)	3.50	3.50	5.00	1.28
E.2 Ordering of lists of options (Not Useful/Useful)	3.80	4.00	5.00	1.24
E.3 Ordering of lists of options (Confusing/Logical)	3.85	4.00	5.00	1.23

Table 6.5: User satisfaction feedback for Section E (List Sorting) (*n*=20)

User satisfaction results from Section F (Map View Adaptation) are summarised in Table 6.6. Visualisation adaptation (in the form of changes to the map-type, zoom-level and location visualised) was rated as useful by most participants in the field study (F.2). Participant opinion regarding how noticeable these changes were (F.1) varied considerably (standard deviation=1.41), indicating fairly large differences in opinion. A review of the distribution of participant responses to F.1 revealed that responses were concentrated at the two extremes of the scale. This indicates that participants were either very aware of the adaptations taking place, or were oblivious to the effects of visualisation adaptation. Several possible explanations exist. MediaMaps provides for browsing media collections in either the list view or the map view. Some participants possibly preferred the list view and therefore did not spend enough time on the map view to notice adaptation taking place. Furthermore, it is possible that the visualisation adaptation (and its benefits) would become more obvious over more prolonged periods of use and that the three week period of the field study was insufficient in this regard. It should also be noted that the visualisation adaptation was designed to be unobtrusive, with changes to visualisation parameters only made when a new visualisation is rendered and not while the user is interacting with the map visualisation (Section 5.9.3.1).

	Mean	Median	Mode	St. Dev.
F.1 Automatic changes to parameters (Not Noticeable/Noticeable)	3.25	4.00	4.00	1.41
F.2 Automatic changes to parameters (Not Useful/Useful)	3.75	4.00	4.00	1.02
F.3 Automatic changes to parameters (Confusing/Logical)	3.65	4.00	4.00	1.09

Table 6.6: User satisfaction results for Section F (Map View Adaptation) (*n*=20)

Correlations were also calculated between the user satisfaction results for the three forms of adaptation (Section D-Section F) and the amount of time participants spent using MediaMaps (Table 6.7).

		Correlation Co-efficient	P-value
Section D	Not Noticeable/Noticeable	0.28	0.26
	Not Useful/Useful	0.31	0.21
	Confusing/Logical	**0.46**	**0.04**
Section E	Not Noticeable/Noticeable	**0.44**	**0.04**
	Not Useful/Useful	**0.62**	**0.01**
	Confusing/Logical	**0.39**	**0.05**
Section F	Not Noticeable/Noticeable	0.09	0.72
	Not Useful/Useful	-0.16	0.53
	Confusing/Logical	-0.19	0.45

Table 6.7: Correlation between time spent using MediaMaps and user satisfaction results for Section D-Section F (Significant correlations shown in bold face, using α=0.05)

These results showed that for information adaptation and interface adaptation, all user satisfaction ratings were positively correlated to the amount of time participants spent using the system. A statistically significant correlation showed that the longer users spent using the system, the more logical the system recommendations regarding the sorting of media items were perceived to be. Interface adaptation was also perceived to be more noticeable, useful and logical the longer participants used MediaMaps (all three of these correlations were statistically significant). All ratings for visualisation adaptation were insignificant. The significant positive correlations observed can be interpreted in two ways. Either participants who were highly satisfied with the adaptations felt motivated to use the system more, or longer periods of system use resulted in greater user satisfaction.

Qualitative feedback from the field study of MediaMaps (including questionnaire results and discussions with participants) revealed that the system was used in a wide variety of situations and contexts. Many participants used the system in different towns and cities and some even in multiple countries during the course of the evaluation. The field study took place during the northern hemisphere summer holidays and as a result, many of the participants used MediaMaps while on holiday and/or travelling. This context of use is ideal for MediaMaps, as users are likely to rely on map-based visualisation of their media collections more when the media are distributed over a larger geographical area, than if most of their media are captured at the same location.

Twelve out of 20 participants (60%) commented favourably on the information adaptation and the support for organising media items into collections based on time and location. Participants were also highly complimentary regarding the ability to location-tag multimedia and view map-based visualisations of their media collections. Given the objectives of this research, the following positive comments regarding information adaptation are of particular interest:

- *"Excellent management capabilities of media items."*
- *"Very good system for organising your digital library, particularly if you are a frequent traveller."*
- *"The sorting multimedia into collections is nice."*

- *"MediaMaps is an excellent idea to sort out all your captured media files per location. It makes it more fun to use and it helps to capture the whole experience."*
- *"This application makes media much easier to find."*
- *"In MediaMaps it is easy to find media based on the location or time"*
- *"...concept of collections and automated decisions is very useful"*
- *"It's a great app for people who do a lot of travelling."*
- *"Location-based collections are useful"*

A common theme was that participants felt that MediaMaps would be particularly useful for organising media captured while travelling and that the automatic organisation and subsequent map-based visualisation make the system an effective tool for relating and reliving their experiences. Many participants also liked the ability to group media items and search by location, supporting previous research (Section 5.2) which identified location as an important variable in helping users locate previously captured photos.

Feedback regarding interface adaptation was also positive, with participants commenting favourably on the easy access to frequently and recently used items. Discussions with users provided further evidence to support the positive quantitative feedback obtained in the user satisfaction questionnaire (Section E) regarding the use of adaptive lists.

Positive feedback was also received regarding the general usability and learnability of MediaMaps. Key themes which were identified are listed in Table 6.8. Participants commented favourably on the ease of use, usefulness, learnability and enjoyable nature of MediaMaps.

	n	**%**	**Example of Actual Comments**
Ease of Use	13	65	*"Very easy to use and intuitive."*
Useful	13	65	*"The application is very useful for viewing media by location."*
Learnability	4	20	*"Intuitive interface, very straight forward to start using."*
Enjoyable	5	25	*"Application was fun to use."*

Table 6.8: Key themes in positive qualitative feedback

The map view of media items/collections was also regarded as useful. Several participants described the map visualisations as easy to use and pleasant to interact with. Positive feedback was also received regarding visualisation adaptation, although one participant commented that they did not notice this form of adaptation as they preferred to use the list view.

Negative feedback largely fell into three categories, as follows:

- *Lack of flexibility in media sorting:* Some participants requested more flexibility regarding the sorting of media items into collections. This included the ability to move items between collections after initial sorting, the ability to change the location associated with an image (in the case of incorrect manual tagging) and the ability to perform manual location-tagging at their own convenience. Some participants had their own preferences regarding how to sort media items. Some preferred only to sort based on location and felt it unnecessary to create multiple collections for the same location on different dates. Many of these issues would be difficult to address without making significant changes to the algorithm used to perform sorting. The real-time, incremental nature of the RED algorithm means that allowing users to perform location-tagging of photos after these photos are captured is not possible without significant modifications to the algorithm. The movement of items from one collection to another could be supported, but would have to be done without affecting algorithm parameters, as in its current form, the algorithm does not cater for adjustments to be made after items have been sorted.

- *Perceived inaccuracies in manual location-tagging:* This issue was addressed previously in this section and was as a result of insufficient place names being stored in the system. This could be addressed by using a reverse geo-coding web service, but this would result in additional data transfer costs (which is why place name tagging was not implemented in this fashion originally).

- *Insufficient use of camera capabilities:* Several participants complained that MediaMaps did not take full advantage of the camera features available on their device. The standard camera application on the Nokia devices used by participants in the field study provides a range of advanced functionality, including auto-flash, digital zooming, colour balance settings and scene modes. Unfortunately, the current

Java implementation on Nokia N-series devices does not implement the camera functionality of the Advanced Multimedia Supplements API. As a result, only basic camera functionality is available to Java applications running on these devices. Furthermore, access to these features would do little to address the objectives of this research. The absence of these features would, however, possibly be a significant barrier to the adoption of MediaMaps as a commercial application.

No negative feedback (aside from criticism regarding the lack of flexibility in media sorting) was received regarding the three forms of adaptation implemented. Few complaints regarding general usability were reported. Those issues that were reported were largely regarding problems inherent to the devices used. For example, on the Nokia N95, the camera is particularly slow, with the device having to be held still for approximately 3 seconds in order to prevent image blur. In addition, opening the camera shutter automatically starts the default camera application (this cannot be disabled) and participants using MediaMaps on a Nokia N95 therefore had to first close this application before continuing to use the system.

A large number of suggestions for improvement were also received, although many of these were either impossible to implement given the restrictions of the development platform used, or had little do to with the objectives of this research.

Suggested improvements were classified into the following categories:

- *Solutions to the negative feedback:* Many of the suggested improvements corresponded directly to the negative feedback received and were discussed above.
- *Improved GPS feedback:* Participants wanted access to more information regarding the available GPS signal, including satellite information (which satellites are being used to pick up their GPS position and associated signal strengths). They also wanted to be able to see from the map view when their GPS position was available. They also wanted their current position clearly shown on the map view (where available) and shown separately from the selection cursor.

- *Sharing media:* Participants wanted the ability to easily share photos captured using MediaMaps from within the system. Options to share via Bluetooth and to upload to photo sharing websites such as Flickr[12] and Picasa[13] were requested.
- *Integration:* Some participants requested that MediaMaps be integrated into the standard Nokia camera application. This is not feasible as a third party developer and therefore not possible using Java.
- *Searching by Place Name:* It was suggested that this process could be streamlined by only including names of places where items have been tagged. This would certainly improve searching by place name and will be integrated into future versions of MediaMaps as an interface adaptation.
- *Additional customisations:* Several improvements were suggested concerning customisations to the file path where media and map tiles are saved. Some participants also wanted to be able to control the photo and video level of detail.

Unlike the positive feedback, very little of the negative feedback and suggested improvements related to the three forms of adaptation implemented in MediaMaps. Nevertheless, many of the improvements discussed above would improve the system and would be useful enhancements if MediaMaps were to be extended.

6.6.8 Summary

Evaluation of the general accuracy of the RED algorithm implemented in MediaMaps revealed that the algorithm performed very well, with accuracy levels approaching 90%. Precision and recall metrics (which focus on the boundaries between different events) were slightly lower, with average precision of 84.81% and average recall of 81.14% achieved. These results are comparable to results published for other algorithms and are particularly encouraging if the low resource requirements and real-time incremental nature of the RED algorithm are considered. False negative errors (where the algorithm failed to detect new events) were found to be the most common. The RED algorithm was also analysed to see whether the algorithm was truly adaptive and displayed improved accuracy in response to

[12] http://www.flickr.com

[13] http://picasaweb.google.com

user feedback. Results showed that this was the case, with a combined plot across all participants revealing a learning trend (Figure 6.5). Location information proved to be useful in ensuring accurate recommendations when a conclusive decision could not be made based on temporal information.

The interface adaptation algorithm also achieved high levels of accuracy. List options were selected from the adaptive section of lists on average 76.78% of the time (100% accuracy is not realistic as the relevant option may not have been recently or frequently selected). This high level of accuracy holds obvious benefits in terms of user efficiency, as scrolling down the list will be unnecessary in cases where the user's desired option is available at the top of the list in question. User satisfaction regarding both information and interface adaptation was shown to be positively correlated to the amount of time participants spent using MediaMaps.

Learnability of MediaMaps was evaluated by examining task times for the most complex system task (capturing, tagging and sorting media items). This revealed that participants were able to complete this task more quickly, the longer they used the system. Qualitative feedback also suggested that participants found MediaMaps easy to use, with several participants noting that the intuitive nature of the system and logical grouping of functions made it easy for first-time users to start using the system.

Quantitative user satisfaction feedback revealed that participants were highly satisfied with their interaction with the system. The high ratings regarding the perceived usefulness of all three forms of adaptation are particularly encouraging given the objectives of this research. Qualitative feedback supported these quantitative results, with information adaptation in particular being highly regarded by the majority of users. The only complaint that was of particular relevance was that participants found the information adaptation algorithm to be lacking in terms of flexibility. Qualitative feedback also revealed that participants were impressed with the general usability of MediaMaps.

In Section 6.3, two high-level evaluation goals were identified. Firstly, the evaluation sought to establish whether the model-based design of adaptive MMV systems based on the Proteus Model is feasible. The implementation of MediaMaps described in the previous chapter and the positive results of this evaluation provide practical and empirical evidence that this is indeed the case. The second high-level objective was to determine whether MediaMaps addresses any of the problems of mobile visualisation and MMV identified in Chapter 2.

The results of this evaluation allow the following benefits to be concluded:

- The accuracy of the information adaptation algorithm, combined with the user satisfaction feedback, suggest that this algorithm supports the tedious task of sorting media items very well. It also suggests that the algorithm allows this task to be performed accurately and effectively. Positive results regarding the ability to locate sorted media items as well as the ability to view map-based visualisations of media collections suggest that the sorting of media items into collections also helped participants to easily retrieve previously captured media items and prevented screen clutter that would otherwise have occurred.
- The accuracy of the interface adaptation suggests that efficiency gains were achieved by allowing participants quick access to their desired option on many occasions. The positive user satisfaction feedback supports this conclusion.
- The nature of the visualisation adaptation algorithm (where the visualisation is adapted according to user behaviour), combined with positive feedback regarding its usefulness suggest that it was useful in tailoring visualisations according to the users' preferences. This is useful in preventing the same zooming, panning and view customisation operations being necessary every time a collection is viewed.
- High ratings in terms of perceived usefulness and usability were achieved for all three forms of adaptation. Qualitative feedback suggests that all three forms of adaptation contributed positively to overall user satisfaction with MediaMaps.

6.7 Conclusions

Field studies were identified as an important tool in the evaluation of both mobile applications and AUIs, as this method allows variables present in the system's actual context of use to be taken into consideration. AUIs are difficult to evaluate for several reasons, including the fact that it is often difficult to attribute usability to the adaptations implemented in the system. Two approaches currently exist to overcome this problem. Firstly, adaptive and non-adaptive versions of the same system can be compared. Secondly, layered evaluations can be conducted, where adaptations are first evaluated regarding how they build up knowledge about the user and then in terms of how they act upon that knowledge. Both of these techniques were considered unsuitable for the evaluation of MediaMaps, due to the unavailability of a non-adaptive version and the nature of the adaptations. It was, therefore, decided to use interaction logging and user satisfaction questionnaires in conjunction with a field study to evaluate the accuracy, perceived usefulness and usability of the three forms of adaptation implemented in MediaMaps.

Preliminary user testing of MediaMaps was conducted prior to the field study. This was done in order to ensure that no significant usability problems existed that would hamper the longer term field study. The effectiveness and user satisfaction of the system was evaluated through the use of questions embedded in a test plan and a user satisfaction questionnaire. Results obtained showed that the system was highly effective in helping participants complete their tasks and high levels of user satisfaction were reported. Several minor usability issues were identified and suggested improvements were incorporated into the system, thereby improving user feedback, map browsing speed and addressing occlusion problems.

An international field study was conducted over three weeks, involving 20 users from four continents. The field study evaluated the usability and learnability of MediaMaps, with a specific focus on the accuracy, perceived usefulness and usability of the three forms of adaptation. This was achieved through the use of interaction logging and user satisfaction questionnaires. In the course of the field study, MediaMaps was used in a wide variety of situations and contexts.

Results showed that general usability and learnability of MediaMaps was very good. The RED information adaptation algorithm achieved average accuracy of 87.44% and precision and recall comparable to other algorithms which use post-processing to cluster photo collections. The RED algorithm was also shown to improve its accuracy as it adapted to user feedback. The adaptive lists achieved average accuracy of 76.78%, which is as high as could be expected given that the user’s desired option cannot always be in the adaptive top section.

Satisfaction results were also encouraging, with participants rating all three forms of adaptation highly in terms of usefulness. Information adaptation, in particular, was regarded as highly useful. Qualitative feedback supported the quantitative questionnaire data. Participants were highly satisfied with the ability to organise their media items into collections based on time and location. The only significant negative feedback received related to the lack of flexibility in terms of the sorting of media items. Location information was shown to be useful for ensuring accurate recommendations regarding the sorting of media items where temporal information was inconclusive. Interface adaptation was also highly regarded, as was the general usability of the system. Many participants described MediaMaps as easy and intuitive to use.

The positive user satisfaction feedback received regarding the adaptations performed suggests that these adaptations were useful in addressing several of the problems associated with MMV. Information adaptation helped users become more effective in sorting, searching and visualising media collections. All three forms of adaptation helped improve user efficiency by minimising the amount of user interaction necessary.

The results of this evaluation provide empirical evidence that the design of an adaptive MMV system based on the Proteus Model is feasible and that adaptation can provide significant benefits for MMV.

In the following chapter, this dissertation is concluded. The contribution and achievements of this research are highlighted. Opportunities for future research are also identified.

Chapter 7: Conclusion

7.1 Introduction

In this chapter, the objectives of this research are revisited in order to examine whether these have been satisfactorily achieved. The theoretical and practical contributions of this research are highlighted and limitations examined. Finally, recommendations for theory, practice and future research are made.

7.2 Achievement of Research Objectives

The primary objective of this research was to develop and evaluate an adaptive MMV system using a model-based design approach (Section 1.3). In order to achieve this goal, the following objectives were identified:

- To identify the problems and shortcomings of existing MMV systems and techniques;
- To determine how AUIs can be used to improve MMV systems;
- To develop a model for the design of adaptive MMV systems;
- To implement a proof-of-concept prototype based on the above model; and
- To determine the benefits provided by this prototype to users.

Two significant benefits of mobile devices as visualisation platforms were identified, namely the ability to deliver visualisations to users on the move and the ability to sense the user's context (Section 2.4.1). Several shortcomings were identified which hamper the design and use of mobile visualisation systems (Section 2.4.2), including limited screen space, limited hardware resources and awkward interaction mechanisms. MMV systems are among the most common mobile visualisation systems, supporting users in accomplishing a wide range of tasks (Section 2.5.2). Many of the problems associated with mobile visualisation are particularly severe in MMV systems (Section 2.5.3) where the large nature of the information space and the complexity of the user's tasks exacerbate the problems of mobile visualisation. Existing techniques for mobile visualisation and MMV were reviewed (Section 2.6). A comparison of the different techniques (Section 2.6.6) showed that existing techniques do not adequately address the problems of MMV and often introduce new problems while solving existing ones. The use of AUIs for MMV systems was identified as a possible solution to the problems and shortcomings of MMV.

The relationship between AUIs and IUIs was examined, with AUIs being identified as a sub-class of IUIs (Section 3.2). A general definition was adopted, which described AUIs as systems which improve interaction with the user by constructing a user model based on previously observed user interaction with the system. AUIs provide a wide range of potential benefits, including helping users to complete complex tasks, assisting them with learning and tailoring information content and presentation to match their preferences (Section 3.4). Several problems and shortcomings of AUIs were identified, including privacy issues, a lack of transparency in the adaptation process and a lack of user control (Section 3.5). AUIs were identified as possessing three high-level components, namely an afferential component for observing and recording user interaction (Section 3.7), an inferential component for inferring user preferences from this interaction data (Section 3.8) and an efferential component for performing adaptation in accordance with user preferences (Section 3.9). Several types of AUIs were identified, with adaptations ranging from information filtering to assistance in completing complex tasks (Section 3.9). Many of these adaptations have the potential to address or alleviate the problems of mobile visualisation and MMV identified in Chapter 2.

Existing literature regarding the use of AUIs in MMV systems was analysed in order to identify the variables which influence adaptation, as well the types of adaptation which are useful for such systems (Section 3.10). Four input variables to the adaptation process were identified, namely users (including their preferences and profile), user tasks, context and devices (Section 3.10.1). In addition to adaptations which cater for device differences, three aspects of MMV systems can be adapted in response to these variables, namely the information visualised, the map-based visualisations and the user interface (Section 3.10.2). Adaptations to the capabilities of different mobile devices were excluded from the scope of this research. Three input variables (user, context and tasks) and adaptation effects (information, visualisation and interface) were therefore considered.

Four existing models related to MMV were examined (Section 3.11). These existing models were analysed in terms of their support for the input variables and adaptation effects discussed above. The UbiquiTO architecture (Section 3.11.1) and the Mobile Cartographic Framework (Section 3.11.3) were identified as providing the most support for these input variables and adaptation effects. Neither the UbiquiTO architecture nor the Mobile Cartographic Framework, however, provide support for all three input variables and adaptation effects. In addition, the UbiquiTO architecture targets the closely related domain of mobile tourist guides and not MMV systems in general. The Mobile Cartographic Framework is conceptual in nature and does not directly support the design of adaptive MMV systems.

The Proteus Model was proposed in Chapter 4, based on the UbiquiTO architecture and the Mobile Cartographic Framework. The Proteus Model addresses the shortcomings of these existing models by supporting adaptation according to users' preferences, tasks and context, while facilitating information, visualisation and interface adaptation. A description of the architecture (Section 4.5), individual components (Section 4.6), interface design (Section 4.7), data design (Section 4.8) and algorithm design (Section 4.9) was included in order to provide a template for the design of adaptive MMV systems.

The implementation of a proof-of-concept prototype based on the Proteus Model was discussed in Chapter 5. This prototype, called MediaMaps, allows the capture, location-tagging, organisation and map-based visualisation of multimedia on mobile phones (Section 5.3). MediaMaps was designed according to the specifications of the Proteus Model. All three forms of adaptation facilitated by the Proteus Model were implemented in MediaMaps. Visualisation adaptation is performed by adapting several parameters of the map-based visualisation of media collections in response to user behaviour, through the use of Naïve Bayesian classifiers (Section 5.8.1). Information adaptation is performed through the automated sorting of media items into collections based on time and location using the RED algorithm (Section 5.8.2). User feedback is used to adapt the algorithm according to user preferences. Interface adaptation is performed through the use of adaptive lists, in which the most recently and frequently selected items are promoted to the top of the list (Section 5.8.3).

MediaMaps was subjected to a preliminary usability evaluation, in which the effectiveness and user satisfaction of the system was evaluated (Section 6.5). The results of the preliminary evaluation revealed that MediaMaps was highly effective in supporting user tasks and high levels of user satisfaction were reported (Section 6.5.7). Several minor usability issues and opportunities for improvement were also identified and resulted in improvements being made to MediaMaps prior to the field study (Section 6.5.8). An international field study was then conducted to evaluate the usability of MediaMaps and the accuracy, usefulness and usability of the three forms of adaptation (Section 6.6).

The field study of MediaMaps involved 20 international participants using the system on their personal mobile phones over a three week period. Interaction logging was built into the system in order to measure the accuracy of the information and interface adaptation algorithms. During the course of the field study, MediaMaps was used in a wide variety of situations and contexts, with several participants using the system while travelling. At the end of the three week period, participants were required to complete a user satisfaction questionnaire to determine the usability and perceived usefulness of the system and the three forms of adaptation. Results obtained revealed that the information and interface adaptations achieved high levels of accuracy (Section 6.6.7). The information adaptation algorithm also demonstrated improvements in accuracy as it learned from user feedback. High levels of user satisfaction were reported, with participants rating all three forms of adaptation as highly useful. Qualitative feedback supported the accuracy information and user satisfaction feedback, with participants regarding the adaptations facilitated by the system as highly useful.

All of the objectives of this research were therefore successfully achieved. In the following section, the theoretical and practical contributions stemming from this research are highlighted.

7.3 Research Contribution

The contribution of this research can be broken down into theoretical and practical contributions, as discussed below.

7.3.1 Theoretical

The Proteus Model (Figure 7.1, repeated here for ease of reference) was proposed in Chapter 4 to address the shortcomings of existing models for the design of adaptive MMV systems. This model included several components drawn from the UbiquiTO architecture, as well as the Mobile Cartographic Framework. Components were also drawn from the fundamental AUI components identified in the literature study of AUIs (Section 3.6).

The Proteus Model provides for adaptation according to users, their context and tasks, through the use of the following models (Section 4.6.2-Section 4.6.5):

- *System Model:* The System Model stores the current status of the adaptable parameters of the system. Changes performed by both the system and the user are reflected in this model in order to maintain a consistent system state.
- *Task Model:* The Task Model allows the system to adapt according to the task the user is currently performing. The task model contains the different user tasks that are possible, decomposed into their constituent sub-tasks. User interaction data can be monitored and this information used in conjunction with the Task Model in order to identify the current user task.
- *User Model:* The User Model stores user preferences, obtained by processing observed user interaction data using a user modelling algorithm.
- *Context Model:* The Context Model stores information about the user's context and the task the user is currently performing. This information is collected through device-based sensors and by monitoring the user's interaction with the system.

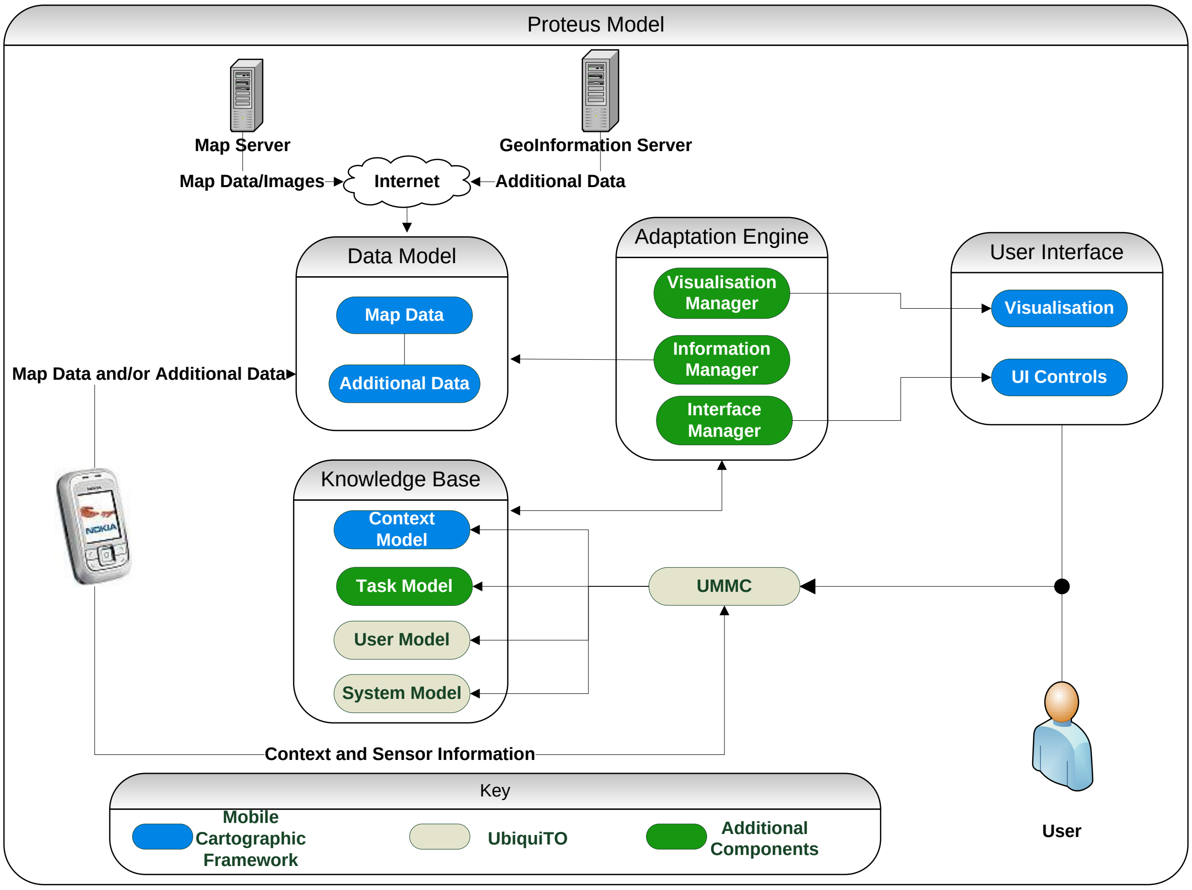

Figure 7.1: The architecture of the Proteus Model

The Proteus Model also caters for all three classes of adaptation effects (information, visualisation and interface adaptation). Individual components handle each form of adaptation, allowing different algorithms to be used for the different forms of adaptation and for these algorithms to be easily substituted with alternatives.

The following three components are responsible for performing adaptation (Section 4.6.6 – Section 4.6.8):

- *Visualisation Manager:* The Visualisation Manager is responsible for adapting the map-based visualisations presented to the user.
- *Information Manager*: The Information Manager is responsible for adapting the data which is visualised, as well as the structure of this data in order to match the user's needs and preferences.
- *Interface Manager*: The Interface Manager is responsible for performing adaptations to the user interface of the MMV system. This could be in terms of the availability or ordering of shortcuts or menu options, the input or output modality and the interaction style (e.g. menus vs. shortcuts).

The final component of the Proteus Model which is involved in performing adaptation is the User Monitoring and Modelling Component (UMMC) (Section 4.6.1). This component combines the monitoring module and user modelling components which are traditionally part of AUIs (Section 3.6) into a single component. The UMMC is responsible for implicitly observing and recording user interaction with the system, processing this interaction data using one or more user modelling algorithms and then storing user preferences in the User Model. The model proposed in Chapter 4 describes the components necessary to perform adaptation and how these components interact with one another.

The Proteus Model represents the first model to incorporate all three classes of adaptation input variables (user, context and tasks) as well as all three classes of adaptation effects (information, visualisation and interface) into a single model. It improves on the conceptual Mobile Cartographic Framework by considering not only user context, but also user behaviour and user tasks. It improves on the UbiquiTO architecture by considering a wider range of input variables and adaptations and also by considering MMV in general as opposed to the narrow domain of mobile tourist guides.

The Proteus Model allows for the design of adaptive MMV systems which harness a wide range of benefits made possible by AUIs. Information adaptation ensures that the information visualised is organised according to the user's needs and that only the relevant information (given the user's task, preferences and context) is provided. This provides the potential to address several of the problems identified in Section 2.4.2 and Section 2.5.3, including the limited screen space and hardware resources, screen clutter and awkward interaction mechanisms. Visualisation adaptation allows the presentation of information to be tailored according to the user's preferences. Optimal use can be made of limited screen space and clutter prevented by rendering only the relevant portion of a visualisation in detail, or displaying legends or keys only when necessary. Both information and visualisation adaptation consider the user's task (e.g. highlighting the recommended route when performing a navigation task). The user's attention (which may be divided between the MMV system and external distractions) can be directed towards the salient information. Interface adaptation can be used to provide easy access to the most relevant system functions. Adaptation of the interaction modality based on the user's context (e.g. read directions aloud while driving) can also provide potential benefits.

Evaluations of AUIs are still rare and fraught with difficulties (Section 6.2.2). Existing approaches, which include comparative evaluation of adaptive and non-adaptive versions and layered evaluation, were considered inappropriate in the evaluation of MediaMaps (Section 6.3). A combination of interaction logging and user satisfaction questionnaires was used to evaluate the adaptations implemented in MediaMaps. A new user satisfaction questionnaire (Appendix F), based on the QUIS, was designed to evaluate the three forms of adaptation implemented. The questionnaire directly addressed the perceived usefulness and usability of the adaptations implemented. Interaction data was used to evaluate the accuracy of the information and interface adaptations implemented.

7.3.2 Practical

A prototype adaptive MMV system, called MediaMaps, was developed as a proof-of-concept of the Proteus Model. MediaMaps is a Java ME application, which allows users to capture, location-tag and visualise multimedia on their mobile phones. MediaMaps was designed using the Proteus Model as a template, thereby demonstrating that the successful implementation of an adaptive MMV system based on the Proteus Model is possible. All the components described in the model were implemented according to specification.

MediaMaps is in itself an original MMV system. No other system exists which allows users to capture, location-tag, sort and view map-based visualisations of multimedia in a single application. Although location-tagging is becoming more common on high-end mobile phones, it is largely limited to photos. In most cases users are required to upload their photos onto the Web in order to visualise their photo collections in a map-based view. The user usually has little control over the appearance of the visualisation and many of these systems are multi-user photo sharing websites, where the user's photo collection is viewable by any user. MediaMaps allows the user's personal media collection to remain private. Users are also able to view their media collections anywhere and at any time. Various options are available to personalise the appearance of the visualisation and the user is able to filter the visualisation according to media type and time period. Searching by time and location can also be performed.

The implementation of the RED algorithm on a mobile device represents the first time that this algorithm has been implemented on an actual mobile device (the original implementation was implemented and tested on a desktop system). The adaptive lists, based on an existing algorithm, also represent one of the first occasions that such an algorithm has been implemented on a mobile device and within a larger application, rather than as part of a limited experimental prototype. MediaMaps also represents one of the first MMV systems to consider all three adaptation input variables (user, context and task) as well as facilitate all three forms of adaptation in a single application.

The evaluation of MediaMaps represents the first evaluation of the RED algorithm on mobile devices. Interaction logs (Section 6.6.7.1) showed that the algorithm achieved very high accuracy levels, approaching 90%. Average precision and recall values (which focus on the boundaries between different events into which media items are sorted), of 84.81% and 81.14% were achieved. These results are particularly significant if compared with existing algorithms which cluster photos into collections using post-processing approaches. These algorithms have been shown to achieve similar precision and recall performance. The results of this evaluation therefore provide evidence to suggest that the RED algorithm is able to achieve accuracy similar to more resource-intensive algorithms and that real-time sorting of media items on mobile phones may be feasible. The results of the evaluation showed that the RED algorithm was prone to false negative errors, rather than false positives. Interaction logging also revealed that the algorithm was able to achieve improved accuracy (learning) in response to user feedback.

The field study of MediaMaps is one of the first occasions that a list-ordering algorithm (such as was used for interface adaptation) has been evaluated as part of the field study of a mobile application. Existing evaluations are generally controlled studies, which evaluate such algorithms in a lab environment, often using simulated mobile environments (e.g. emulators) rather than actual mobile devices. Long-term evaluations that consider context of use variables are rarely conducted. Analysis of the interaction data logged during the field study showed that average accuracy of 76.78% was attained. Given that the user's desired option will not always be among the most recently and frequently selected options, this accuracy level is very encouraging, as efficiency benefits can result from users not having to scroll and search for their desired option.

Feedback from the user satisfaction questionnaire was very positive, with high levels of user satisfaction reported. Analysis of the questionnaire feedback showed that the participants regarded all three forms of adaptation as highly useful. The information adaptation (recommendations regarding the sorting of media items into collections) was rated particularly highly. Qualitative feedback supported the quantitative data, with the majority of the participants providing positive feedback regarding the ability to sort and visualise their media collections based on time and location. The usability of all three forms of adaptation was assessed by asking users whether they noticed adaptation taking place and whether the adaptations performed were logical. Feedback received showed that the participants found the adaptations to be logical and unobtrusive. Participants were also asked whether the popup messages providing system recommendations and requesting feedback were annoying. User feedback suggested that these messages were rated as being more annoying than was the case in the preliminary usability evaluation, suggesting that a less intrusive means of incorporating feedback needs to be devised. Qualitative feedback regarding the general usability and learnability of MediaMaps was also very positive. The only significant negative qualitative feedback received pertained to the information adaptation, where users requested more flexibility in the way media items are sorted.

The implementation and evaluation of MediaMaps demonstrated that the model-based design of adaptive MMV systems is possible. The adaptations which were implemented address many of the problems of mobile visualisation and MMV identified in Section 2.4.2 and Section 2.5.3. The information adaptation in MediaMaps prevents users from having to use the awkward interaction mechanisms provided by mobile phones to sort media items into collections. Visualisation of media collections (rather than all of the individual media items) prevents visualisations from becoming cluttered. The visualisation adaptations performed allow users to view visualisations that match their preferences without having to repeatedly perform customisation. The visualisation and information adaptations implemented in MediaMaps also ensure that only information relevant to the user is visualised. The interface adaptation (ordering of list options) also means that less user interaction is required in order to find desired options.

7.4 Limitations and Problems Encountered

Several problems were encountered during the implementation and evaluation of MediaMaps. The limitations of the available implementation platforms meant that it was not possible to integrate the system with the standard camera application on the Nokia N-Series devices used for evaluation. This became problematic on devices such as the Nokia N95, where the default camera application is activated when the camera lens cover is opened. Several problems were also encountered during the field study, resulting in the reduction of the initial sample size from 26 to 20. Two participants encountered technical problems with their phones during the course of the evaluation and as a result had to send their phones in for repair. One participant updated the firmware on their mobile device during the field study and as a result was unable to supply the necessary interaction log files. Three participants did not complete the field study satisfactorily, as a review of their logged interaction data revealed very limited use of the system.

The following limitations of this research can be identified:

- The scope of this research was limited to exclude adaptation to the capabilities of different mobile devices. The Proteus Model does not support this, due to the additional complexities of achieving this form of adaptation. Such adaptation has been widely addressed in existing work and generally requires consideration of very low-level implementation issues. The results are also often of limited value.
- The Proteus Model has only been tested through the implementation of a single prototype, namely MediaMaps. In order to more thoroughly test the model and to prove its versatility, more adaptive MMV systems need to be designed and implemented based on the model. More evaluations are also needed in order to determine whether the model-based design of adaptive MMV systems allows the problems associated with these systems to be addressed.

- The evaluation of MediaMaps did not directly compare an adaptive and a non-adaptive version, due to the fact that adaptation was built into the functionality of the system, making any comparison with a non-adaptive version infeasible. As a result, it is difficult to provide conclusive proof that the adaptations performed improved the usability of the system. Nevertheless, our evaluation results (in which the adaptations were directly evaluated) yielded promising results regarded the users' perceptions of the adaptations performed. Any comparative evaluation would probably also require a larger sample size in order to produce statistically significant results.

7.5 Recommendations

Recommendations resulting from this research can be divided into recommendations for theory, practice and future research, as discussed below.

7.5.1 Recommendations for Theory

The successful model-based design and implementation of MediaMaps (based on the Proteus Model) provides practical evidence that model-based design of adaptive MMV systems is feasible.

The results of the evaluation of MediaMaps provide supporting evidence for several existing theories. It has been suggested on several occasions that AUIs can provide a means of addressing the problems of MMV (Section 3.10), with little empirical evidence to support this claim. The results of the field study show that an AUI incorporating accurate adaptations (which users perceived as useful and usable) was developed. The accuracy of the adaptations also implies efficiency gains for users. These findings provide empirical evidence to support other studies suggesting that AUIs provide benefits for improving MMV systems.

Existing literature states that users regard location as an important variable when trying to find previously captured photos (Section 5.2). Qualitative and quantitative evidence collected during the evaluation of MediaMaps showed that this was indeed the case, with users regarding the ability to sort and search their media collections by time and location as very useful. Field study feedback suggested that users also regarded the ability to view map-based visualisations of their media collections on their mobile phones as useful, particularly for media captured while travelling. The results of the field study also showed that when temporal information was inconclusive, location information helped to ensure that accurate recommendations were provided regarding the sorting of media items into collections.

Evaluations of AUIs are still rare (Section 6.2.2). This can be ascribed to the difficulties associated with the evaluation of AUIs and the inadequacies of existing evaluation methods. In the evaluation of MediaMaps, the existing approaches of comparative evaluation and layered evaluation were not considered appropriate. Instead, interaction logging and user satisfaction questionnaires were used in the evaluation of the prototype. This technique allowed the accuracy of the information and interface adaptations to be evaluated through user feedback. It also allowed correlations between actual usage data and user perceptions (obtained using the questionnaire) to be calculated in order to determine whether adaptation accuracy and system usage patterns impacted on user satisfaction. A user satisfaction questionnaire based on the QUIS (Appendix F) was developed to evaluate the usability and usefulness of the adaptations implemented in MediaMaps.

7.5.2 Recommendations for Practice

The Proteus Model provides a detailed template for the design and implementation of adaptive MMV systems incorporating a wide range of possible adaptations. The model is detailed enough to guide the design and implementation of new adaptive MMV systems, while still being general enough to support a wide range of potential applications. Model-based design provides many advantages, including reduced risk and managed complexity, with positive results more likely if a tested design (such as that of the Proteus Model) is employed. The Proteus Model is therefore of potential benefit to designers of future adaptive MMV systems. This includes research prototypes, where adaptation can be implemented in a modular fashion with different implementations of the model components used to compare different approaches and adaptation techniques.

7.5.3 Recommendations for Future Research

Further research is also needed to investigate how MMV systems can adapt in order to better match the user’s needs. Investigation is needed to determine how the different tasks facilitated by MMV systems (Section 2.5.2) can be supported through adaptations to the information, visualisation and user interface. More research is also needed to investigate how existing techniques which have been developed to perform MMV using non-adaptive means (Section 2.6) can be improved through the use of adaptation (Section 3.10). Combining visualisation techniques which cater for the constraints of mobile devices with techniques which adapt to the behaviour and preferences of the individual user could provide a successful approach to dealing with the problems and shortcomings of MMV.

Several extensions to the Proteus Model are possible. The model could be extended to allow MMV systems to adapt to different mobile devices (this could be integrated both as an input to the adaptation process and as an adaptation effect). Provision could be made to integrate adaptive services, which provide relevant information and functionality given the user's behaviour, context and tasks. The model could be adjusted to cater for similar domains (e.g. mobile tourist guides such as the UbiquiTO system, Section 3.11.1). Considering that many of the problems of MMV are also encountered by mobile visualisation in general, the Proteus Model could also be generalised to cater for adaptive mobile visualisation.

Additional MMV systems need to be developed to further test and refine the Proteus Model. While the development and evaluation of MediaMaps showed that the model can be used for the model-based design and development of an adaptive MMV system, it does not demonstrate whether the model is flexible enough to cater for a wide range of systems within this domain. The implementation (as well as evaluation) of additional prototypes would allow us to conclude with a greater degree of confidence that the Proteus Model does allow a wide range of adaptive MMV systems to be developed which provide benefits to the user and address the problems of MMV. Comparative studies of the design and implementation of the same MMV system with and without the Proteus Model would allow us to determine whether the model delivers the theoretical benefits of model-based design. Comparative evaluation of adaptive and non-adaptive versions of the same application would also provide more conclusive evidence as to whether the model facilitates adaptations which provide measurable usability benefits. The flexibility of the Proteus Model could also be evaluated by using different implementations of the same component, in order to compare different adaptation techniques or user modelling algorithms.

The evaluation of MediaMaps (Section 6.5 and Section 6.6) revealed several possibilities for extending the system and for future research. Comparative evaluations between the RED algorithm used for information adaptation and other algorithms for clustering photos (Section 5.8.2) could be conducted. The resource requirements, accuracy, precision and recall of the RED algorithm could be compared to those of existing algorithms using either existing test photo collections, or user photo collections collected through a field study. The RED algorithm could be modified to provide for more flexibility in terms of the sorting of media items into collections, as requested by participants during the field study. Interface adaptation could also be subjected to a more detailed evaluation. Different list ordering algorithms

(Section 3.9.1) could be compared, either in controlled studies or in long-term field studies. Interface adaptation could also be extended to include shortcuts, menu options and input or output modalities. Visualisation adaptation in MediaMaps could be enhanced through the use of vector maps, rather than raster maps. This would make a greater degree of adaptation possible, including adaptation in terms of the map features which are displayed, the level of detail, the map colours and map symbols used (amongst others). A longer field study of MediaMaps could be conducted, to determine whether user perceptions of the system and the adaptations change over a longer period of use. MediaMaps could also be transformed from a single user application to a multi-user photo-sharing platform. This would provide possibilities for collaborative user modelling techniques (Section 3.8) to be used to adapt visualisation parameters or recommend locations that users are likely to be interested in photographing.

References

Ahmed, S. and Ashraf, G. (2006): Model-based User Interface Engineering with Design Patterns. *The Journal of Systems and Software*, **60** (8):1408-1422. Elsevier Science Inc.

Al-bar, A. and Wakeman, I. (2001): A Survey of Adaptive Applications in Mobile Computing. In *Proceedings of International Conference on Distributed Computing Systems (ICDCSW '01)*, Arizona, USA:246-251. April 16-19, 2001. IEEE Computer Society.

Alimohideen, J., Renambot, L., Leigh, J., Johnson, A., Grossman, R.L. and Sabala, M. (2006): PAVIS: Pervasive Adaptive Visualization and Interaction Service. In *Proceedings of CHI 2006 Workshop on Information Visualization and Interaction Techniques for Collaboration Across Multiple Displays*, Montreal, Canada.

Alvarez-Cortes, V., Zayas-Perez, B. E., Zarate-Silva, V. H. and Ramirez Uresti, J. A. (2007): Current Trends in Adaptive User Interfaces: Challenges and Applications. In *Proceedings of the Electronics, Robotics and Automotive Mechanics Conference*, Washington, D.C., USA:312-317. September 25-28. IEEE Computer Society.

Andriakopoulou, E., Garofalakis, J. and Plessas, A. (2007): A Mobile Web Map Application for Route Discovery with Device Independent Characteristics. In *Proceedings of Euro American Conference On Telematics And Information Systems (EATIS) 2007*, Faro, Portugal:1-6. May 14-17, 2007. ACM Press.

Apple (2008): *Apple iPhone* [online]. Available at www.apple.com/iphone. [Accessed on 24 November 2008].

Ardissono, L., Goy, A., Petrone, G., Segnan, M. and Torasso, P. (2003): Intrigue: Personalized Recommendation of Tourist Attractions for Desktop and Hand-held Devices. *Applied Artificial Intelligence*, **17** (8-9):687-714. Taylor and Francis Ltd.

Aris, A., Gemmell, J. and Lueder, R. (2004): *Exploiting Location and Time for Photo Search and Storytelling in MyLifeBits*. Technical Report MSR-TR-2004-102. Microsoft Research.

Barnum, C.M. (2002): *Usability Testing and Research*. Needham Heights, MA, USA, Longman Publishers.

Baudisch, P. (2004): Halo: A Virtual Periphery for Small Screen Devices. In *Proceedings of International Conference on Advanced Visual Interfaces (AVI '04)*, Gallipoli, Italy:80-84. May 25, 2004. Springer.

Baudisch, P., Good, N., Bellotti, V. and Schraedley, P. (2002): Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 2002)*, Minneapolis, Minnesota, USA:259-266. April 20-25, 2002. ACM Press.

Baudisch, P. and Rosenholtz, R. (2003): Halo: A Technique for Visualizing Off-Screen Locations. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 2003)*, Fort Lauderdale, Florida, USA:481-488. ACM Press.

Baus, J., Chreverst, K. and Kray, C. (2005): A Survey of Mobile Tourist Guides. *Map-based Mobile Services: Theories, Methods and Implementations*:197-216. Springer-Verlag.

Bederson, B., Clamage, A., Czerwinski, M. and Robertson, G. (2004): DateLens: A Fisheye Calendar Interface for PDAs. *ACM Transactions on Computer-Human Interaction*, **11** (1):90-119. March 2004. ACM Press.

Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D. and Furnas, G. (1996): Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, **7**:3-31.

Betts, B.J., Mah, R.W., Papasin, R., Del Mundo, R., McIntosh, D.M. and Jorgensen, C. (2005): *Improving Situational Awareness for First Responders via Mobile Computing*. NASA/TM-2005-213470. Smart Systems Research Laboratory, NASA Ames Research Center.

Billsus, D., Brunk, C., Evans, C., Gladish, B. and Pazzani, M. (2002): Adaptive Interfaces for Ubiquitous Web Access. *Communications of the ACM*, **45** (5):34-38. May 2002. ACM Press.

Björk, S., Redström, J., Ljungstrand, P. and Holmquist, L (2000): Powerview: Using Information Links and Information Views to Navigate and Visualize Information on Small Displays. In *Proceedings of the 2nd international Symposium on Handheld and Ubiquitous Computing*, Bristol, UK:46-62. September 25 - 27, 2000. Springer-Verlag.

Bornträger, C., Cheverst, K., Davies, N., Dix, A., Friday, A. and Seitz, J. (2003): Experiments with Multi-Modal Interfaces in a Context-Aware City Guide. *Lecture Notes in Computer Science 2795 (Mobile HCI 2003)*:116-130. Springer.

Boström, F., Floréen, P., Liu, T., Nurmi, P., Oikarinen, T-K., Vetek, A. and Boda, P. (2008a): Capricorn - An Intelligent User Interface for Mobile Widgets. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '08)*, Gran Canaria, Spain:417-418. ACM Press.

Boström, F., Floréen, P., Liu, T., Nurmi, P., Oikarinen, T-K., Vetek, A. and Boda, P. (2008b): Capricorn - An Intelligent User Interface for Mobile Widgets. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2008)*, Amsterdam, the Netherlands:327-330. September 2-5, 2008. ACM Press.

Bridle, R. and McCreath, E. (2006): Inducing Shortcuts on a Mobile Phone Interface. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '06)*, Sydney, Australia:327-329. ACM Press.

Browne, D., Norman, M. and Riches, D. (1990): Why Build Adaptive Systems? In *Adaptive User Interfaces*:16-57. D. Browne, P. Totterdell and M. Norman (Eds.) London, UK, Academic Press.

Brusilovsky, P., Karagiannidis, C. and Sampson, D. (2001): The Benefits of Layered Evaluation of Adaptive Applications and Services. In *Proceedings of Eighth International Conference on User Modelling (UM 2001)*, Freiburg, Germany:1-8.

Brusilovsky, P. and Schwarz, E. (1997): User as Student: Towards an Adaptive Interface for Advanced Web-Based Applications. In *Proceedings of Sixth International Conference on User Modeling (UM '97)*, Chia Laguna, Sardinia, Italy:177-188. June 2-5 1997. Springer.

Burigat, S. and Chittaro, L. (2005): Visualizing the Results of Interactive Queries for Geographic Data on Mobile Devices. In *Proceedings of Symposium on Advances in Geographic Information Systems (GIS 2005)*, Bremen, Germany:277-284. ACM Press.

Burigat, S. and Chittaro, L. (2008): Interactive Visual Analysis of Geographic Data on Mobile Devices based on Dynamic Queries. *Journal of Visual Languages and Computing*, **19** (1):99-122. February 2008. Academic Press, Inc.

Burigat, S., Chittaro, L. and Gabrielli, S. (2006): Visualizing Locations of Off-Screen Objects on Mobile Devices: A Comparative Evaluation of Three Approaches. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '06)*, Helsinki, Finland:239-246. September 12-15, 2006. ACM Press.

Burigat, S., Chittaro, L. and Parlato, E. (2008): Map, Diagram and Web Page Navigation on Mobile Devices: The Effectiveness of Zoomable User Interfaces with Overviews. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2008)*, Amsterdam, the Netherlands:147-156. ACM Press.

Card, S., Mackinlay, J. and Shneiderman, B. (1999): *Readings in Information Visualization: Using Vision to Think*. San Francisco, USA, Morgan Kaufmann.

Carmo, M.B., Afonso, A.P., Pombinho de Matos, P. and Vaz, A. (2008): MoViSys – A Visualization System for Geo-Referenced Information on Mobile Devices. *Lecture Notes in Computer Science 5188 (Visual 2008)*:167-178. Springer-Verlag.

Cena, F., Console, L., Gena, C., Goy, A., Levi, G., Modeo, S. and Torre, I. (2006): Integrating Heterogeneous Adaptation Techniques to Build a Flexible and Usable Mobile Tourist Guide. *AI Communications*, **19** (4):369-384. IOS Press.

Chalmers, D., Sloman, M. and Dulay, N. (2001): Map Adaptation for Users of Mobile Systems. In *Proceedings of World Wide Web Conference (WWW 2001)*, Hong Kong:735-744. May 2001. ACM Press.

Chen, W. and Chen, M. (2007): Event detection in real time on a personal camera phone image collection. *Multimedia Systems* (12):479-492. 22 February 2007. Springer.

Chi, E.H. and Riedl, J.T. (1998): An Operator Interaction Framework for Visualization Systems. In *Proceedings of Symposium on Information Visualization (InfoVis '98)*, Research Triangle Park, North Carolina, USA:63-70. IEEE Computer Society.

Chin, D.N. (2001): Empirical Evaluation of User Models and User-Adapted Systems. *User Modelling and User Adapted Interaction* (11):181-194.

Chin, J.P., Diehl, V.A. and Norman, K.L. (1988): Development of an Instrument Measuring User Satisfaction of the Human Computer Interface. In *Proceedings of SIGCHI '88*, New York:213-218. ACM Press.

Chittaro, L. (2004): HCI Aspects of Mobile Devices and Services. *Personal and Ubiquitous Computing*, **8** (2):69-70. Springer-Verlag.

Chittaro, L. (2006): Visualizing Information on Mobile Devices. *IEEE Computer*, **39** (3):40-45. March 2006. IEEE Computer Society.

Cockburn, A. and Savage, J. (2003): Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. In *Proceedings of People and Computers XVII: British Computer Society Conference on Human Computer Interaction*, Bath, England:87-102.

Collins (1991): *Collins Concise Dictionary*. Glasgow, Scotland, William Collins & Sons Ltd.

Comscore (2008): *M:Metrics* [online]. Available at http://www.comscore.com/press/release.asp?press=2321. [Accessed on 24 November 2008].

Cooper, M., Foote, J., Girgensohn, A. and Wilcox, L. (2005): Temporal Event Clustering for Digital Photo Collections. *ACM Transactions on Multimedia Computing, Communications and Applications*, **1** (3):269-288. ACM Press.

Crow, D. and Smith, B. (1993): The Role of Built-In Knowledge in Adaptive Interface Systems. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '93)*, Orlando, Florida, USA:97-104. January 4-7, 1993. ACM Press.

Davis, F.D. (1989): Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, **13**:319-340.

De Bra, P., *et al.* (2003): AHA! The Adaptive Hypermedia Architecture. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia (HT'03)*, Nottingham, United Kingdom:81-84. August 26 - 30, 2003. ACM Press.

de la Flor, G. (2004): User Modeling and Adaptive User Interfaces, University of Bristol. Available at http://www.ilrt.bris.ac.uk/publications/researchreport/rr1085. [Accessed on 24 November 2008].

Dey, A. K., Salber, D.A. and Abowd, G. D. (2001): A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction 2001*, **16** (2-4):97-166.

Dieterich, H., Malinowski, U., Kuhme, T. and Schneider-Hufschmidt, M. (1993): State of the Art in Adaptive User Interfaces. In *Adaptive User Interfaces: Principles and Practice*:13-48. M. Schneider-Hufschmidt, T. Kuhme and U. Malinowski (Eds.) Amsterdam, the Netherlands, Elsevier Science Publishers.

Domik, G.O. and Gutkauf, B. (1994): User Modeling for Adaptive Visualization Systems. In *Proceedings of IEEE Conference on Visualization '94*, Washington D.C., USA:217-223. October 17-21, 1994. IEEE Computer Society.

Dumas, J.S. and Redish, J.C. (1999): *A Practical Guide to Usability Testing*. Norwood, NJ, USA, Ablex Publishing Corp.

Ehlert, P. (2003): *Intelligent User Interfaces: Introduction and Survey*. Report DKS03-01 / ICE 01. Department of Information Technology and Systems, Delft University of Technology. Delft, the Netherlands.

Eisenstein, J. and Rich, C. (2002): Agents and GUIs From Task Models. In *Proceedings of Intelligent User Interfaces (IUI '02)*, San Francisco, California, USA:47-54. January 13-16, 2002. ACM Press.

Eklund, J. and Sinclair, K. (2000): An Empirical Appraisal of the Effectiveness of Adaptive Interfaces for Instructional Systems. *Educational Technology and Society*, **3** (4):165-177.

Elkan, C. (1997): *Naive Bayesian Learning*. Technical Report CS97-557. Department of Computer Science and Engineering, University of California, San Diego.

Engelbrecht, A.P. (2002): *Computational Intelligence*. Chichester, UK, 1st Edn, John Wiley & Sons Ltd.

FCC (2003): *Enhanced 911* [online]. Available at http://www.fcc.gov/pshs/services/911-services/enhanced911/Welcome.html. [Accessed on 24 November 2008].

Findlater, L. and McGrenere, J. (2008): Impact of Screen Size on Performance, Awareness and User Satisfaction With Adaptive Graphical User Interfaces. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*, Florence, Italy:1247-1256. April 5-10. ACM.

Fink, J. and Kobsa, A. (2002): User Modeling for Personalized City Tours. *Artificial Intelligence Review*, **18** (1):33-74. Kluwer Academic Publisher.

Fischer, G. (2001): User Modeling in Human-Computer Interaction. *User Modeling and User Adapted Interaction (UMUAI)*, **11** (1-2):65-86.

Flickr (2008): *Flickr.com* [online]. Available at www.flickr.com. [Accessed on 24 November 2008].

Fresco, R., Marucci, L. and Signore, O. (2003): Adaptive Hypermedia for User-Centred Geovisualization in Multiplatform Environments. In *Proceedings of ACM Conference on Hypertext and Hypermedia (HyperText '03)*, Nottingham, UK:20-21. August 26-30, 2003. ACM Press.

Fuchs, G., Luboschik, M., Hartmann, K., Ali, K., Schumann, H. and Strothotte, T. (2006): Adaptive Labeling for Interactive Mobile Information Systems. In *Proceedings of Information Visualization (IV'06)*, Washington, D.C, USA:453-459. July 5-7, 2006. IEEE Computer Society.

Fuller, J. (2006): *Virtual Earth Mobile 1.64* [online]. Available at http://blogs.msdn.com/windowsmobile/archive/2006/06/27/649384.aspx. [Accessed on 24 November 2008].

Furnas, G. (1986): The FISHEYE view: a new look at structured files. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, Boston, Massachussets, USA:16-23. ACM.

Gajos, K.Z., Czerwinski, M., Tan, D.S. and Weld, D.S. (2006): Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of Working Conference for Advanced Visual Interfaces (AVI'06)*, Venezia, Italy:201-208. May 23 - 26, 2006. ACM Press.

Gartner (2008): *Gartner Says Worldwide Smartphone Sales Grew 16 Percent in Second Quarter of 2008* [online]. Available at http://www.gartner.com/it/page.jsp?id=754112. [Accessed on 24 November 2008].

Gena, C. (2005): Methods and Techniques for the Evaluation of User-Adaptive Systems. *The Knowledge Engineering Review*, **20** (1):1-37. Cambridge University Press.

Gomaa, H. and Hussein, M. (2007): Model-Based Software Design and Adaptation. In *Proceedings of International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*:7-16. IEEE Computer Society.

Greenberg, S. and Witten, I. (1985): Adaptive personalized interfaces: A question of viability. *Behaviour and Information Technology*, **4** (1):31-45.

Gutwin, C. and Fedak, C. (2004): Interacting with Big Interfaces on Small Screens: a Comparison of Fisheye, Zoom, and Panning Techniques. In *Proceedings of 2004 Conference on Graphics Interface*, London, Ontario, Canada:145-152. May 17-19, 2004. Canadian Human-Computer Communications Society.

Hakala, T., Lehikoinen, J. and Aaltonen, A. (2005): Spatial Interactive Visualization on Small Screen. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria:137-144. September 19-22, 2005. ACM Press.

Hampe, M. and Paelke, V. (2005): Adaptive Maps for Mobile Applications. Mobile Maps 2005: Interactivity and Usability of Map-based Mobile Services. Workshop at Mobile HCI 2005, 19 September 2005, Salzburg, Austria.

Han, J. and Kamber, M. (2006): *Data Mining Concepts and Techniques*. San Francisco, California, USA, 2nd Edn, Morgan Kaufmann Publishers.

Hang, A., Rukzio, E. and Greaves, A. (2008): Projector Phone: A Study of Using Mobile Phones with Integrated Projector for Interacting with Maps. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2008)*, Amsterdam, the Netherlands:207-216. September 2-5, 2008. ACM Press.

Hao, J. and Zhang, K. (2007): A Mobile Interface for Hierarchical Information Visualization and Navigation. In *Proceedings of IEEE International Symposium on Consumer Electronics*:1-7. IEEE Computer Society.

Harada, S., Naaman, M., Song, Y.J., Wang, Q. and Paepcke, A. (2004): Lost in Memories: Interacting With Photo Collections On PDAs. In *Proceedings of the Joint ACM/IEEE Conference on Digital Libraries (JCDL'04)*, Tucson, Arizona, USA:325-333. June 7-11, 2004. ACM.

Harrie, L., Sarjakoski, T. and Lehto, L. (2002): A Variable-Scale Map for Small Display Cartography. In *Proceedings of Joint International Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada:8-12. July 8-12, 2002.

Hedberg, S.R. (1998): Is AI going mainstream at last? A look inside Microsoft Research. *IEEE Intelligent Systems*:21-25. March/April, 1998. IEEE Computer Society.

Heer, J. (2004): *Visualizing Online Social Networks* [online]. Available at http://www.cs.berkeley.edu/~jheer/vizster/early_design/. [Accessed on 24 November 2008].

Hefley, W.E. and Murray, D. (1993): Intelligent User Interfaces. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '93)*, Orlando, Florida, USA:3-10. January 4-7. ACM Press.

Höök, K. (1999): Designing and Evaluating Intelligent User Interfaces. In *Proceedings of the International Conference on Intelligent User interfaces (IUI '99)*, Los Angeles, California, USA:5-6. January 05 - 08, 1999. ACM Press.

Hui, B. and Boutilier, C. (2006): Who's Asking for Help? A Bayesian Approach to Intelligent Assistance. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '06)*, Sydney, Australia:186-193. January 29 - February 1, 2006. ACM Press.

ISO9241-11 (1998): Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability. ISO.

Jameson, A. (2003): Adaptive Interfaces and Agents. In *Human-Computer Interaction Handbook*:305-330. A.J. Jacko and A. Sears (Eds.) Mahwah, USA, Erlbaum.

Jones, M. and Marsden, G. (2005): *Mobile Interaction Design*. Chichester, UK, John Wiley & Sons Ltd.

Jones, S., Jones, M., Marsden, G., Patel, D. and Cockburn, A. (2005): An Evaluation of Integrated Zooming and Scrolling on Small Screens. *International Journal of Human-Computer Studies* (65):271-303. Elsevier.

Kaikkonen, A., Kallio, T., Kekäläinen, A., Kankainen, A. and Cankar, M. (2005): Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *Journal of Usability Studies*, **1** (1):4-16. UPA.

Kaptelinin, V. (1995): A Comparison of Four Navigation Techniques in a 2D Browsing Task. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 1995)*, Denver, Colorado, USA:282-283. May 7-11, 2005. ACM Press.

Karagiannidis, C., Koumpis, A. and Stephanidis, C. (1995): Supporting Adaptivity in Intelligent User Interfaces: The case of Media and Modalities Allocation. In *Proceedings of ERCIM Working Group on User Interfaces for All Workshop "Towards User Interfaces for All: Current Efforts and Future Trends"*, Heraklion, Greece:30-31. October 30-31, 1995.

Karstens, B., Kreuseler, M. and Schumann, H. (2003): Visualization of Complex Structures on Mobile Handhelds. In *Proceedings of International Workshop on Mobile Computing*.

King, J.L. and West, J. (2002): Ma Bell's orphan: US cellular telephony, 1947–1996. *Telecommunications Policy* (26):189-203.

Kjeldskov, J. and Graham, C. (2003): A Review of Mobile HCI Research Methods. In *Proceedings of Mobile HCI 2003 (Lecture Notes in Computer Science 2795)*, Udine, Italy:317-335. Springer-Verlag.

Knapp, L. (1995): A Task Analysis Approach to the Visualisation of Geographic Data. *Cognitive Aspects of Human Computer Interaction for Geographic Information Systems, Behavioural and Social Sciences*:355-371. Kluwer.

Kobsa, A. (2004): Adaptive Interfaces. In *Encyclopedia of Human-Computer Interaction*. W.S. Bainbridge (Ed.), Berkshire Publishing Group, Great Barrington, MA.

Kobsa, A., Koenemann, J. and Pohl, W. (2001): Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *The Knowledge Engineering Review*, **16** (2):111-155.

Koelle, D. (2004): *Intelligent User Interfaces* [online]. Available at http://www.davekoelle.com/intint.jsp. [Accessed on 24 November 2008].

Kofod-Petersen, A. and Aamodt, A. (2003): Case-Based Situation Assessment in a Mobile Context-Aware System. In *Proceedings of the Conference on Artificial Intelligence in Mobile Systems (AIMS)*. Available at http://w5.cs.uni-sb.de/~krueger/aims2003/camera-ready/kofod-petersen-6.pdf. [Accessed on 24 November 2008].

Korvemaker, B. and Greiner, R. (1999): *The Trials and Tribulations of Building an Adaptive User Interface* [online]. Available at http://www.cs.ualberta.ca/~greiner/PAPERS/AdaptUI.pdf. [Accessed on 24 November 2008].

Kraak, M-J. (2006): Beyond Geovisualization. *IEEE Computer Graphics and Applications*:6-9. July/August 2006. IEEE Computer Society.

Krogsæter, M., Opperman, R. and Thomas, C. (1994): A User Interface Integrating Adaptability and Adaptivity. In *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*:97-125. Hillsdale, New Jersey, USA, Lawrence Erlbaum Associates.

Krogsæter, M. and Thomas, C. (1994): Adaptivity: System-Initiated Individualization. In *Adaptive User Support*:67-96. R. Opperman (Ed.) Hillsdale, New Jersey, USA, Lawrence Erlbaum Associates.

Kühme, T. (1993): A User-Centered Approach to Adaptive Interfaces. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '93)*, Orlando, Florida, USA:243-245. January 4 - 7, 1993. ACM Press.

Kules, B. (2000): *User Modeling for Adaptive and Adaptable Software Systems* [online]. Available at http://www.otal.umd.edu/uuguide/wmk/. [Accessed on 24 November 2008].

Kumar, H., Plaisant, C. and Shneiderman, B. (1997): Browsing Hierarchical Data with Multi-level Dynamic Queries and Pruning. *International Journal of Human-Computer Studies*, **46** (1):103-124. Elsevier.

Lacaze, X. and Palanque, P. (2004): Comprehensive Handling of Temporal Issues in Task Models: What is needed and How to Support it? In *Proceedings of Workshop on the Temporal Aspects of Work for HCI at CHI 2004*, Vienna, Austria. May, 2004.

Langley, P. (1999): User Modeling in Adaptive Interfaces. In *Proceedings of Seventh International Conference on User Modeling*, Banff, Alberta, Canada:357-370. June 20-24, 1999. Springer.

Lennard, A. and Parkes, A. (1995): *An Architecture for Adaptive Interfaces*. AAI/AI-ED Technical Report No.111. Computing Department, Lancaster University. Lancaster, UK.

Looije, R., te Brake, G. and Neerincx, M. (2007): Usability Engineering for Mobile Maps. In *Proceedings of International Conference on Mobile Technology, Applications, and Systems (Mobility '07)*, Singapore:532-539. 10-12 September 2007.

Loui, A. and Savakis, A. (2000): Automatic Image Event Segmentation and Quality Screening for Albuming Applications. In *Proceedings of IEEE International Conference on Multimedia and Expo*:1125-1128. IEEE Computer Society.

MacEachren, A. and Kraak, M-J. (2001): Research Challenges in Geovisualization. *Cartography and Geographic Information Science*, **28** (1):3-12.

Masthoff, J. (2002): The Evaluation of Adaptive Systems. In *Adaptive Evolutionary Information Systems*:329-347. N.V. Patel (Ed.), Idea Group Publishing.

Maybury, M.T. (1998): Intelligent User Interfaces: An Introduction. In *Readings in Intelligent User Interfaces*:1-13. M.T. Maybury and W. Wahlster (Eds.) San Francisco, Morgan Kaufmann.

Microsoft (2008): *Microsoft Live Maps* [online]. Available at http://maps.live.com. [Accessed on 24 November 2008].

Mitchell, J. and Shneiderman, B. (1989): Dynamic versus static menus: an exploratory comparison. *SIGCHI Bulletin*, **20** (4):33-37. April 1989.

Mitchell, K., Race, N.J.P. and Clarke, M. (2005): CANVIS: Context-Aware Network Visualization using Smartphones. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, Salzburg, Austria:175-182. September 19-22. ACM Press.

Mitrovic, N., Royo, J. and Mena, E. (2007): Performance Analysis of an Adaptive User Interface System Based on Mobile Agents. In *Proceedings of Engineering Interactive Systems '07*

(Lecture Notes in Computer Science 4940), Salamanca, Spain. March 22-24, 2007. Springer Verlag.

Mob4Hire (2008): *Mob4Hire - Crowdsourced Mobile Application Testing* [online]. Available at www.mob4hire.com. [Accessed on 24 November 2008].

Mohagheghi, P. and Aagedal, J. (2007): Evaluating Quality in Model-Driven Engineering. In *Proceedings of International Workshop on Modeling in Software Engineering (MISE)*:6-11. IEEE Computer Society.

Murray (1987): Embedded User Models. In H.-J. Bullinger and B. Shackel (Eds.)*Proceedings of Human-Computer Interaction: INTERACT '87*:229-235. Elsevier.

Naaman, M., Jiun Song, Y., Paepcke, A. and Garcia-Molina, H. (2004): Automatic Organization for Digital Photographs with Geographic Coordinates. In *Proceedings of CM/IEEE-CS Joint Conference on Digital Libraries (JCDL '04)*, Tucson, Arizona, USA:53-62. June 7-11, 2004. ACM.

Naaman, M., Nair, R. and Kaplun, V. (2008): Photos on the Go: A Mobile Application Case Study. In *Proceedings of SIGHCI Conference on Human Factors in Computing Systems (CHI '08)*, Florence, Italy:1739-1748. April 5-10, 2008. ACM.

Nielsen, J. (1994a): *Heuristics for User Interface Design* [online]. Available at http://www.useit.com/papers/heuristic/heuristic_list.html. [Accessed on 24 November 2008].

Nielsen, J. (1994b): *Usability Engineering*. Boston, Academic Press.

NIST (2001): *Common Industry Format for Usability Test Reports* [online]. Available at www.nist.gov/iusr. [Accessed on 24 November 2008].

Nivala, A-M. and Sarjakoski, L. (2005): Adapting Map Symbols for Mobile Users. In *Proceedings of International Cartographic Conference (ICC '05)*, A Coruna, Spain. July 9-16.

Nokia (2008a): *Nokia Location Tagger* [online]. Available at www.nokia.com/betalabs/locationtagger. [Accessed on 24 November 2008].

Nokia (2008b): *Nokia Maps* [online]. Available at http://europe.nokia.com/A4509271. [Accessed on 24 November 2008].

Nurmi, P. and Hassinen, M. (2007): A Comparative Analysis of Personalization Techniques for a Mobile Application. In *Proceedings of International Conference on Advanced information Networking and Applications Workshops (AINAW '07)*, **2**:270-275. May 21-23, 2007. IEEE Computer Society.

O'Grady, M.J. and O'Hare, G.M.P. (2004): Enabling Customized and Personalized Interfaces in Mobile Computing. In *Proceedings of the International Conference on intelligent User interfaces (IUI'04)*, Madeira, Portugal:292-294. January 13 - 16, 2004. ACM.

Olivier, M. (2004): *Information Technology Research: A practical guide for Computer Science and Informatics*. Pretoria, 2nd Edn, Van Schaik Publishers.

Openwave (2002): *Overview of Location Technologies* [online]. Available at http://developer.openwave.com/omdtdocs/location_studio_sdk/pdf/Intro_to_Location_Technologies.pdf. [Accessed on 24 November 2008].

Opperman, R. (1994): *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale, New Jersey, USA, Lawrence Erlbaum Associates.

Papadogiorgaki, M., Papastathis, V., Nidelkou, E., Kompatsiaris, I., Waddington, S., Bratu, B. and Ribiere, M. (2007): Distributed User Modeling for Personalized News Delivery in Mobile Devices. In *Proceedings of Second International Workshop on Semantic Media Adaptation and Personalization*:80-85. IEEE.

Paramythis, A., Totter.A. and Stephanidis, C. (2001): A Modular Approach to the Evaluation of Adaptive User Interfaces. In S. Weibelzahl, D. Chin and G. Weber (Eds.)*Proceedings of Empirical evaluation of adaptive systems: Proceedings of workshop at the Eighth International Conference on User Modeling*, Freiburg, Germany:9-24. July 13, 2001.

Paternò, F and Mancini, C. (2000): Model-Based Design of Interactive Applications. *Intelligence*, **11** (4):26-38. December 2000.

Pattath, A., Bue, B., Jang, Y., Ebert, D., Zhong, X., Ault, A. and Coyle, E. (2006): Interactive Visualization and Analysis of Network and Sensor Data on Mobile Devices. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, Baltimore, Maryland, USA:83-90. 31 October - 2 November. IEEE Computer Society.

Pauty, J., Couderc, P. and Banâtre, M (2005): Using Context to Navigate Through a Photo Collection. In *Proceedings of International Conference on Human Computer Interaction with Mobile Devices & Services (Mobile HCI '05)*, Salzburg, Austria, **111**:145-152. September 19 - 22, 2005. ACM.

Paymans, T.F., Lindenberg, J. and Neerincx, M. (2004): Usability Trade-offs for Adaptive User Interfaces: Ease of Use and Learnability. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '04)*, Madeira, Portugal:301-303. January 13-16, 2004. ACM.

Plaisant, C., Milash, B., Rose, A., Widoff, S. and Shneiderman, B. (1996): LifeLines: Visualizing Personal Histories. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*, Vancouver, Canada:221-227. April 13-18. ACM.

Platt, J.C., Czerwinski, M. and Field, B.A. (2003): PhotoTOC: Automatic Clustering for Browsing Personal Photographs. In *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing*, Singapore, **1**:6-10. 15-18 December 2003.

Preece, J., Rogers, Y. and Sharp, H. (2007): *Interaction Design: beyond human-computer interaction*. 2nd Edn, John Wiley & Sons.

Quesenbery, W. (2003): The Five Dimensions of Usability. *Content and Complexity: Information Design in Technical Communication*. Lawrence Erlbaum Associates.

Rao, B. and Minakakis, L. (2003): Evolution of Mobile Location-based Services. *Communications of the ACM*, **46** (12):61-65. ACM.

Rao, R. and Card, S. (1994): The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*:318-322. ACM.

Reichenbacher, T. (2003): Adaptive Methods for Mobile Cartography. In *Proceedings of 21st International Cartographics Conference*, Durban, South Africa:1311-1322. August 10-16.

Reichenbacher, T. (2004): *Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices* Institut für Photogrammetrie und Kartographie, Technischen Universitat, Munchen. Munich, Germany.

Reigelsberger, J. and Nahimovsky, Y. (2008): Seeing the Bigger Picture: A Multi-Method Field Trial of Google Maps for Mobile. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*, Florence, Italy:2221-2228. April 5-10. ACM.

Rhyne, T.-M. and MacEachren, A. (2004): Visualizing Geospatial Data. *ACM SIGGRAPH Course Notes*, ACM. [Accessed on August 8-12, 2004].

Ricci, F. and Nguyen, Q.N. (2007): Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. *IEEE Intelligent Systems*, **22** (3):22-29. IEEE Computer Society.

Robbins, D., Cutrell, E., Sarin, R. and Horvitz, E. (2004): ZoneZoom: Map Navigation for Smartphones with Recursive View Segmentation. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI '04)*, Gallipoli, Italy:231-234. May 25, 2004. ACM.

Rogers, S., Fiechter, C-N. and Langley, P. (1999): An Adaptive Interactive Agent for Route Advice. In *Proceedings of Third International Conference on Autonomous Agents*, Seattle, USA:198-205. ACM.

Rosenbaum, R. and Schumann, H. (2005): Grid-based Interaction for Effective Image Browsing on Mobile Devices. In *Proceedings of SPIE*, **5684**:170-180. SPIE - The International Society for Optical Engineering.

Ross, E. (2000): *Intelligent User Interfaces: Survey and Research Directions*. Report CSTR-00-004. Department of Computer Science, University of Bristol. Available at www.cs.bris.ac.uk/Publications/Papers/1000447.pdf. [Accessed on 24 November 2008].

Rumbaugh, J., Jacobson, I. and Booch, G. (2004): *Unified Modeling Language Reference Manual*. 2nd Edn, Pearson Higher Education.

Sanna, A. and Fornaro, A. (2003): IMoViS: A System for Mobile Visualization of Intrusion Detection Data. *Information & Security.*, **12** (2):235-249.

Schmidt, A., Beigl, M. and Gellersen, H-W. (1999): There is More to Context than Location. *Computer and Graphics*, **6** (21):893-901. Elsevier.

Sears, A. and Shneiderman, B. (1994): Split menus: effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, **1** (1):27-51. March 1994. ACM Press.

Segal, R. and Kephart, J. (1999): MailCat: An Intelligent Assistant for Organizing E-Mail. In *Proceedings of Third International Conference on Autonomous Agents*:276-282. ACM Press.

Shneiderman, B. (1996): The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, Los Alamos, California, USA:336-343. September 1996. IEEE Computer Society.

Skupin, A. (2000): From Metaphor to Method: Cartographic Perspectives on Information Visualization. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis '00)*, Salt Lake City, Utah, USA:91-97. October 9-10, 2000. IEEE Computer Society.

Sommerville, I. (2006): *Software Engineering*. 8th Edn, Addison-Wesley.

Spence, R. (2007): *Information Visualization: Design for Interaction*. 2nd Edn, Pearson Education Limited.

Spenceley, S. and Warren, J. (1998): The Intelligent Interface for Online Electronic Medical Records using Temporal Data Mining. *the Thirty-First Hawaii International Conference on Systems Sciences*:266-274. IEEE Computer Society.

Stone, D., Jarrett, C., Woodroffe, M. and Minocha, S. (2005): *User Interface Design and Evaluation*. San Francisco, California, USA, Morgan Kaufmann.

Strachan, L., Anderson, J., Sneesby, M. and Evans, M. (1997): Pragmatic User Modelling in a Commercial Software System. In *Proceedings of the Sixth International Conference on User Modelling*, Sardinia, Italy:189-200. June 2-5 1997. Springer.

Strachan, L., Anderson, J., Sneesby, M. and Evans, M. (2000): Minimalist User Modelling in a Complex Commercial Software System. *User Modeling and User-Adapted Interaction*, **10** (2-3):109-146.

Tamminen, S., Oulasvirta, A., Toiskallio, K. and Kankainen, A. (2004): Understanding Mobile Contexts. *Journal of Personal and Ubiquitous Computing* (8):135-143.

Telaphia (2007): *Q1 2007 Mobile LBS Overview* [online]. Available at http://developer.navteq.com/getDocument.do?docId=1543. [Accessed on 24 November 2008].

Totterdell, P. and Rautenbach, P. (1990): Adaptation as a problem of design. In *Adaptive User Interfaces*:59-84. D. Browne, P. Totterdell and M. Norman (Eds.) London, Academic Press.

Tsandilas, T. and Schraefel, M.C. (2004): Usable Adaptive Hypermedia Systems. *New Review of Hypermedia and Multimedia*, **10** (1):5-29.

Tullis, T. and Albert, B. (2008): *Measuring the User Experience*. Morgan Kaufmann.

Van den Bergh, J. and Coninx, K. (2004): Model-Based Design of Context-Sensitive Interactive Applications: A Discussion of Notations. In *Proceedings of the Annual Conference on Task*

Models and Diagrams (TAMODIA '04), Prague, Czech Republic, **86**:43-50. November 15-16. ACM.

van Tonder, B. and Wesson, J. (2008): Using Adaptive Interfaces to Improve Mobile Map-based Visualisation. In C. Cilliers, L. Barnard and R. Botha (Eds.)*Proceedings of SAICSIT '08*, Wilderness, South Africa:257-266. 6-8 October 2008. ACM.

van Velsen, L.S., van der Geest, T.M. and Klaassen, R.F. (2006): User-centred Evaluation of Adaptive and Adaptable Systems. In *Proceedings of the 5th Workshop on User-Centered Design and Evaluation of Adaptive Systems at Adaptive Hypermedia '06*, Dublin, Ireland.

van Velsen, L.S., van der Geest, T.M. and Klaassen, R.F. (2007): Testing the Usability of a Personalized System: Comparing the Use of Interviews, Questionnaires and Thinking-Aloud. In *Proceedings of IEEE International Professional Communication Conference (IPCC '07)*:1-8. 1-3 October 2007. IEEE Computer Society.

Vasilyeva, E., Pechenizkiy, M. and Puuronen, S. (2005): Towards the Framework of Adaptive User Interfaces for eHealth. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*:139-144. June 23 - 24, 2005. IEEE Computer Society.

Von Hunolstein, S. and Zipf, A. (2003): Towards Task-Oriented Map-based Mobile Guides. In *Proceedings of International Workshop "HCI in Mobile Guides", Mobile HCI '03*, Undine, Italy. 8-11 September 2003.

Wackerly, D., Mendenhall, W. and Scheaffer, R. (2002): Mathematical Statistics with Applications.536-539. Pacific Grove, California, USA, Duxbury.

Wang, Y., Liu, X., Chen, X., Chen, Y. and Meng, L. (2001): Adaptive Geovisualization - an Approach towards the Design of Intelligent Geovisualization Systems. *Journal of Geographical Sciences*, **11**:1-8.

Weibelzahl, S. and Weber, G. (2002): Advantages, Opportunities and Limits of Empirical Evaluation: Evaluating Adaptive Systems. *Künstliche Intelligenz*:17-20.

Whitten, J., Bentley, L. and Dittman, K. (2004): *Systems Analysis and Design Methods*. New York, 6th Edn, McGraw-Hill/Irwin.

Yoo, J., Gervasio, M. and Langley, P. (2003): An Adaptive Stock Tracker for Personalized Trading Advice. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI' 03)*, Miami, Florida, USA:197-203. January 12-15, 2003. ACM.

Zellweger, P., Mackinlay, J., Good, L., Stefik.M. and Baudisch, P. (2003): City Lights: Contextual Views In Minimal Space. In *Proceedings of CHI '03 Extended Abstracts on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA:838-839. April 05 - 10, 2003. ACM.

Zhang, D., Karabatis, G., Chen, Z., Adipat, B., Dai, L., Zhang, Z. and Wang, Y. (2006): Personalization and Visualization on Handheld Devices. In *Proceedings of the ACM Symposium on Applied Computing (SAC '06)*, Dijon, France:1008-1012. April 23-27. ACM.

Zipf, A. (2002): User-Adaptive Maps for Location-Based Services (LBS) for Tourism. In K. Woeber, A. Frew and M. Hitz (Eds.)*Proceedings of the 9th International Conference for Information and Communication Technologies in Tourism (ENTER 2002)*, Innsbruck, Austria:329-338. Springer-Verlag.

Zipf, A. and Neis, P. (2007): Realizing Focus Maps with Landmarks using OpenLS Services. In *Proceedings of LBS and Telecartography 2007*, Hong Kong.

Zipf, A. and Richter, K-F. (2002): Using Focus Maps to Ease Map Reading: Developing Smart Applications for Mobile Devices. *Künstliche Intelligenz (KI) (Artificial Intelligence). Special Issue: Spatial Cognition.* (4):35-37.

Zukerman, I. and Albrecht, D. (2001): Predictive Statistical Models for User Modeling. *User Modeling and User-Adapted Interaction (UMUAI)*, **11** (1-2):5-18.

Appendix A: Preliminary Evaluation Test Plan

Nelson Mandela Metropolitan University, Department of Computer Science and Information Systems

This test plan is part of research towards a MSc in Computer Science and Information Systems

MediaMaps Preliminary Evaluation

Thank you for participating in this evaluation of the *MediaMaps* system.

Overview:

MediaMaps is an application which allows you to capture multimedia (photos, videos and sound recordings) and automatically sorts these media items into collections based on when and where they were captured. These collections can then be browsed in a map-based view, showing the location where the different collections and individual media items were captured. Searching by date and location is also provided in order to help you locate previously recorded multimedia quickly and easily.

Please complete the following tasks, and complete the questionnaire that will be provided once all tasks have been completed.

1. Start *MediaMaps* by selecting the *MediaMaps* icon from the Application menu.
2. Select the media collection named “Humewood Jun 2008”
    a. Select the option to *View on Map*.
    b. Scroll around the map and locate all the media items.
    c. Zoom out so that all media items are visible.
    d. How many media items are there in this collection?

    ____

3. Switch to Animated View and browse through the entire collection.
4. Customise the *Settings*.
    a. Switch back to *Static View*.

b. Change the map type to *Hybrid*.
c. Change the media type to *Photos and Sound*.
d. How many media items are now visible?

____

5. Search by *Date*
    a. Search for a collection created on 9 June 2008.
    b. Select any collection created on this date.
    c. What is the name of the collection you selected?

    ____

    d. How many photos are in this collection?

    ____

6. Search by L*ocation*
    a. Search for a collection in Central.
    b. Select any collection in this area.
    c. What is the name of the collection you selected?

    ____

    d. What is the name of the most recently captured item in this collection?

    ____

7. Capture and save at least 5 photos and 2 videos
    a. Select C*apture* from the main menu and select the appropriate media type.
    b. Capture the photo or video.
    c. If no GPS position is available, the system will request that you select your position from the map.
    d. Find your approximate position and select *Save* or press the 'fire' key.
    e. The system will then present its suggestion regarding whether the new media item belongs in the current media collection, or whether it belongs in a new collection. Select Y*es* or N*o* to confirm or correct the system's suggestion.
    f. What is the name of the collection the first photo was added to?

    ____

8. Find and view the first photo you took
    a. Select the first collection you created in task 7.
    b. Select the photos sub-folder.
    c. Locate the first photo you took.
    d. What is the name of this photo?

____

9. Find and view the first video you took
    a. Select a collection you created in task 7 which contains the first video you recorded.
    b. Select the option to *View on Map.*
    c. Locate the first video you recorded.
    d. What is the name of this video?

    ____

Appendix B: Preliminary Evaluation User Satisfaction Questionnaire

Nelson Mandela Metropolitan University

Department of Computer Science and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Questionnaire: MediaMaps

MediaMaps allows you to capture multimedia (photos, videos and sound recordings) and automatically sorts these media items into collections based on when and where they were captured. These collections can then be browsed in a map-based view, showing the location where the different collections and individual media items were captured. Searching by date and location is also provided in order to help you locate previously recorded multimedia quickly and easily.

	Section A: Biographical Details (mark with X where appropriate)					
1	Gender:	Male	Female			
2	Age:					
3	Occupation:					
4	Mobile phone experience (years)	< 1	1 - 2	2 - 4	> 4	
5	Camera phone experience (years)	< 1	1 - 2	2 - 4	> 4	N/A
6	How often do you use your phone to capture photos/videos/sound	Never	About once a month	About once a week	A few times a week	Every day

	Section B: Interface Design					
1	Were the screen layouts clear?	Never				Always
		1	2	3	4	5
2	Sequence of screens	Confusing				Clear
		1	2	3	4	5
3	Was the user interaction obvious?	Not at All				Very Much
		1	2	3	4	5
4	Icon used to indicate media collections	Confusing				Clear
		1	2	3	4	5
5	Icons used to indicate media items (photos, videos and sound)	Confusing				Clear
		1	2	3	4	5
6	Understanding time visualisation (when media/collection was captured)	Difficult				Easy
		1	2	3	4	5

	Section C: Navigation and Browsing					
1	Menu navigation	Difficult				Easy
		1	2	3	4	5
2	Map manipulation (scrolling, zooming)	Difficult				Easy
		1	2	3	4	5
3	Browsing collection in map view	Difficult				Easy
		1	2	3	4	5
4	Navigating animated view	Difficult				Easy
		1	2	3	4	5
5	Browsing collection in non-map view	Difficult				Easy
		1	2	3	4	5
	Section D: Searching for Media					
1	Searching by date	Difficult				Easy
		1	2	3	4	5
2	Searching by location	Difficult				Easy
		1	2	3	4	5
3	Locating captured media	Difficult				Easy
		1	2	3	4	5
	Section E: Capturing Media					
1	Capturing photo	Difficult				Easy
		1	2	3	4	5
2	Capturing video	Difficult				Easy
		1	2	3	4	5
3	Selecting location manually (if applicable)	Difficult				Easy
		1	2	3	4	5
	Section F: Sorting Media into Collections					
1	Sorting of items into collections	Useless				Useful
		1	2	3	4	5
2	Sorting of items into collections	Complicated				Simple
		1	2	3	4	5
3	Confirmation messages regarding which collection an item belongs in	Annoying				Not Annoying
		1	2	3	4	5
4	Automatic collection naming	Useless				Useful
		1	2	3	4	5
5	Locating media collections	Difficult				Easy
		1	2	3	4	5

6	Locating media items	Difficult				Easy
		1	2	3	4	5
7	Support for managing media items/collections	Useless				Useful
		1	2	3	4	5
8	Browsing media collections in a map view	Useless				Useful
		1	2	3	4	5
9	Collection naming preference (select appropriate option)					
	Automatic naming	Manual naming	Make automatic names default, but allow me to edit			
	Section G: General					
1	Describe positive aspects of the system					
2	Describe negative aspects of the system					
3	Please provide any general comments or suggestions for improvement below					

Appendix C: Preliminary Evaluation Test Procedure

The test procedure for the preliminary evaluation of MediaMaps is provided below:

- The test participant was greeted by the test administrator (the author) on arrival.
- The test administrator provided the participant with the Nokia N95 with MediaMaps installed and gave the participant a brief overview of how to operate the phone. The participant was given an opportunity to get accustomed to operating the phone and ask the test administrator any questions regarding its operation.
- The test administrator then gave the participant an overview of MediaMaps, describing its basic functionality. The shortcut to launch the application was also pointed out.
- The test participant was then provided with the test plan and post-test questionnaire and the purpose of these documents was explained. The questions included in the test plan were pointed out and their purpose explained.
- The test participant then commenced with the evaluation, following the test plan. The test administrator observed the test participant throughout the evaluation, noting any comments made or problems encountered.
- Once the participant had completed the test plan, he/she was required to complete the post-test questionnaire.
- Once the questionnaire was completed, the test administrator briefly interviewed the participant regarding any issues that arose during the test. The participant was encouraged to provide comments or suggestions for improvement and to discuss specific aspects of MediaMaps. The administrator was also available to answer any participant questions.
- The participant was thanked for his/her participation in the evaluation.
- Data from the questionnaire and test plans was captured and analysed.

Appendix D: Preliminary Evaluation User Satisfaction Results

Question	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Mean	Median	Std. Dev.
B.1	4	5	3	5	5	4	4	5	5	4	4.40	4.50	0.70
B.2	4	5	3	5	5	5	4	5	4	2	4.20	4.50	1.03
B.3	4	4	3	4	5	4	4	4	4	4	4.00	4.00	0.47
B.4	5	4	5	5	4	5	5	5	5	4	4.70	5.00	0.48
B.5	5	4	5	5	4	5	5	5	2	3	4.30	5.00	1.06
B.6	4	3	4	5	4	3	3	5	2	4	3.70	4.00	0.95
C.1	4	5	4	5	5	3	5	5	4	4	4.40	4.50	0.70
C.2	4	4	4	4	5	4	4	5	5	3	4.20	4.00	0.63
C.3	4	3	5	4	4	4	5	5	5	4	4.30	4.00	0.67
C.4	5	4	3	4	5	5	5	5	5	2	4.30	5.00	1.06
C.5	4	4	3	5	5	5	5	5	4	3	4.30	4.50	0.82
D.1	5	5	3	5	5	5	5	5	5	4	4.70	5.00	0.67
D.2	5	4	4	4	5	4	5	4	5	4	4.40	4.00	0.52
D.3	5	3	5	3	5	4	5	5	5	4	4.40	5.00	0.84
E.1	5	4	5	5	5	5	5	5	5	4	4.80	5.00	0.42
E.2	5	4	5	5	5	5	5	5	4	4	4.70	5.00	0.48
E.3	4	5	4	4	5	4	5	3	4	4	4.20	4.00	0.63
F.1	4	4	3	5	5	4	4	5	5	5	4.40	4.50	0.70
F.2	4	5	2	5	5	4	5	5	4	4	4.30	4.50	0.95
F.3	4	2	4	5	5	5	5	5	5	4	4.40	5.00	0.97
F.4	4	4	3	5	5	5	4	5	5	4	4.40	4.50	0.70
F.5	4	4	4	4	5	5	4	5	5	4	4.40	4.00	0.52
F.6	4	5	4	4	5	5	4	5	5	4	4.50	4.50	0.53
F.7	2	3	4	5	5	4	5	5	5	3	4.10	4.50	1.10
F.8	4	3	5	4	5	3	5	4	3	3	3.90	4.00	0.88

Table D.1: Results from preliminary evaluation user satisfaction questionnaire (*n*=10)

Appendix E: Field Study Test Plan

Nelson Mandela Metropolitan University
Department of Computer Science and Information Systems

This test plan is part of research towards a MSc in Computer Science and Information Systems

MediaMaps Field Study Test Plan

Thank you for participating in this evaluation of the MediaMaps system.

Overview:

MediaMaps is a Java ME application which allows you to capture and location-tag multimedia (photos, videos and sound recordings) and automatically sorts these media items into collections based on when and where they were captured. These collections can then be browsed in a map-based view on your phone, showing the location where the different collections and individual media items were captured. Searching by date and location is also provided in order to help you locate previously recorded multimedia quickly and easily.

Important:

Before using the system, it is important that you set the following permissions in Application Manager for MediaMaps to "Always Allowed". This will allow you to use the system without constant warning messages when it needs to write to or read from files. To do so, go to Application Manager and select the "MediaMaps.jad" (installed) option. Select open, and then change the following permissions:

- Multimedia (for capturing of multimedia)
- Read user data (in order to create log files and other system files)
- Write user data (in order to read the above files)
- Positioning (in order to use GPS)

Also, do NOT use the default camera application to capture multimedia. With the Nokia N95 this starts by default when you open the lens cover. Please close this application when it starts up.

Objectives:

MediaMaps is designed to adapt to how you use it. You will be asked to evaluate these adaptations after you have finished using the system, making this the most important part of the evaluation. The system adapts in the following ways:

1. Sorting multimedia into collections
    a. The system attempts to help you sort media items into collections, looking at when and where they were recorded in order to group them into collections of related items.
    b. Feedback you give the system, regarding whether captured multimedia belongs in a new or an existing collection, affects future recommendations the system makes.
2. Ordering of lists
    a. The top three items of lists of options (such as lists of collections) contain the most recently and most frequently selected options, allowing items you are likely to select to be accessed more quickly.
3. Adapting map view
    a. Parameters such as the initial zoom-level, map-type, data types, location and time period being visualised are adjusted, based on previous behaviour when viewing the currently selected collection.

General Guidelines:

The evaluation of MediaMaps will last for three weeks. During this period, you need to use the system as often as possible, around five times a week (for at least 10 minutes at a time). It is important that MediaMaps be used in as natural a fashion as possible, recording and browsing multimedia collections. At the end of the evaluation period, you will be required to complete a short questionnaire (including providing gender and age range) and to make available a log file of your use which will be automatically recorded on your phone. DO NOT DELETE THE SYSTEM AND ITS ASSOCIATED FILES UNTIL YOU HAVE EMAILED THIS FILE TO THE ADDRESS PROVIDED IN THE QUESTIONNAIRE. The evaluation is primarily a usability evaluation. We are interested in your opinion of the system and the adaptive functionality.

Tasks:

The following tasks are guidelines regarding the functionality of MediaMaps you need to make use of over the three week period of the evaluation. These tasks need to be performed repeatedly.

1. Capture Media (Photos, Videos and Sound). Use MediaMaps as your primary multimedia capturing software for the duration of the evaluation. Important: do **not** use the standard camera application (which on the Nokia N95 starts automatically when opening the lens cover).
2. Browse Media Collections. Once you've captured a few media items (and as you continue to do so), you can browse your media collections by selecting one from either the list of collections, or from the map provided by MediaMaps.
3. Search for Media. To locate particular collections or media items, search for them, either by date, or by selecting the name of the place they were recorded.
4. Customise Settings. MediaMaps provides the capability to select the map type you would like to view and to customise the media types being visualised. Change these settings as required.

Appendix F: Field Study User Satisfaction Questionnaire

Nelson Mandela Metropolitan University

Department of Computer Science and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Questionnaire: MediaMaps

Thank you for participating in this evaluation of MediaMaps. Complete the questionnaire and click the "Submit by Email" button at the bottom (or save and email to csbbpvt@nmmu.ac.za). If you wish to provide additional feedback beyond this questionnaire, please feel free to do so by email. **Please also email the log file "Decisions.xml" to be found at c:\images\models\decisions.xml on your phone and all interaction files (e.g. Interaction1.xml) in c:\images\interactiondata\.** MediaMaps allows you to capture and location-tag multimedia (photos, videos and sound recordings) and automatically sorts these media items into collections based on when and where they were captured. These collections can then be browsed in a map-based view, showing the location where the different collections and individual media items were captured. Searching by date and location is also provided in order to help you locate previously recorded multimedia quickly and easily. The goal of this questionnaire is to evaluate the system and how it adapts. The system adapts in the following ways:

- Sorting multimedia into collections ·
- Ordering of lists ·
- Adapting the map view

	Section A: Biographical Details (mark with X where appropriate)					
1	Gender:	Male	Female			
2	Age:	< 20	20-29	30-39	40-49	>=50
3	Occupation:					
4	Mobile phone experience (years)	< 1	1 - 2	3 - 4	>= 5	
5	Camera phone experience (years)	< 1	1 - 2	3 - 4	>= 5	
6	Mobile phone skill level	Novice	Intermediate	Expert		
7	How often do you use your phone to capture photos/videos/sound	Never	About once a month	About once a week	A few times a week	Every day

	Section B: Capturing Media					
1	Capturing photos	Difficult				Easy
		1	2	3	4	5
2	Capturing videos	Difficult				Easy
		1	2	3	4	5

3	Selecting location manually	Difficult				Easy
		1	2	3	4	5
	Section C: Searching for Media					
1	Search by date	Difficult				Easy
		1	2	3	4	5
2	Search by place name	Difficult				Easy
		1	2	3	4	5
3	Locating captured media	Difficult				Easy
		1	2	3	4	5
	Section D: Sorting Media into Collections					
1	Sorting of items into collections	Not useful				Useful
		1	2	3	4	5
2	Sorting of items into collections	Difficult				Easy
		1	2	3	4	5
3	Confirmation messages regarding which collection an item belongs in	Annoying				Not Annoying
		1	2	3	4	5
4	Automatic collection naming	Not Useful				Useful
		1	2	3	4	5
5	Locating media collections	Difficult				Easy
		1	2	3	4	5
6	Locating media items	Difficult				Easy
		1	2	3	4	5
7	Support for managing media items/collections	Not Useful				Useful
		1	2	3	4	5
8	Browsing media collections in map view	Not Useful				Useful
		1	2	3	4	5
9	Sorting of items into collections	Not noticeable				Noticeable
		1	2	3	4	5
10	Sorting of items into collections	Confusing				Logical
		1	2	3	4	5
	Section E: List Sorting					
1	Ordering of lists of options	Not Noticeable				Noticeable
		1	2	3	4	5
2	Ordering of lists of options	Not Useful				Useful
		1	2	3	4	5
3	Ordering of lists of options	Confusing				Logical
		1	2	3	4	5

	Section F: Sorting Media into Collections					
1	Automatic changes to map view parameters	Not Noticeable				Noticeable
		1	2	3	4	5
2	Automatic changes to map view parameters	Not Useful				Useful
		1	2	3	4	5
3	Automatic changes to map view parameters	Confusing				Logical
		1	2	3	4	5
	Section G: General					
1	Describe positive aspects of the system					
2	Describe negative aspects of the system					
3	Please provide any general comments or suggestions for improvement below					

Appendix G: Field Study User Satisfaction Results

Question	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10	Participant 11	Participant 12	Participant 13	Participant 14	Participant 15	Participant 16	Participant 17	Participant 18	Participant 19	Participant 20	Mean	Median	Mode	St. Dev.
B.1	5	5	2	2	5	5	4	4	4	2	4	5	5	4	4	3	5	4	5	4	4.05	4.00	5.00	1.05
B.2	5	5	2	2	5	5	4	4	4	2	4	4	5	4	4	3	5	3	4	4	3.90	4.00	4.00	1.02
B.3	4	2	4	5	2	3	3	1	3	1	3	4	3	4	3	1	5	5	3	2	3.05	3.00	3.00	1.28
C.1	5	3	5	4	5	4	4	3	4	2	5	4	2	5	5	2	5	4	5	4	4.00	4.00	5.00	1.08
C.2	3	5	5	4	5	3	3	3	4	2	5	4	2	5	5	4	2	2	4	4	3.70	4.00	5.00	1.13
C.3	3	5	5	4	5	4	3	1	5	2	4	3	4	5	5	5	4	4	4	5	4.00	4.00	5.00	1.12
D.1	4	5	3	5	5	4	2	5	5	3	5	4	5	5	5	3	3	4	5	4	4.20	4.50	5.00	0.95
D.2	4	3	2	4	2	4	2	2	5	3	5	4	3	5	5	3	4	4	4	5	3.65	4.00	4.00	1.09
D.3	5	4	1	3	4	4	3	4	5	3	4	3	4	4	3	1	3	2	3	5	3.40	3.50	4.00	1.14
D.4	5	5	1	2	3	4	5	4	5	3	5	3	4	5	5	4	4	4	3	2	3.80	4.00	5.00	1.20
D.5	4	5	3	4	4	5	5	1	5	3	4	4	4	5	5	4	3	4	5	4	4.05	4.00	4.00	1.00
D.6	4	5	3	4	5	3	4	2	5	3	5	4	3	4	5	4	4	4	5	4	4.00	4.00	4.00	0.86
D.7	5	3	3	4	5	4	4	3	4	4	5	4	2	4	5	3	5	3	4	2	3.80	4.00	4.00	0.95
D.8	5	5	5	5	5	4	4	4	5	4	4	4	3	2	5	4	2	5	5	5	4.25	4.50	5.00	0.97
D.9	5	4	3	3	4	3	3	3	2	3	5	3	5	3	5	3	5	1	4	4	3.55	3.00	3.00	1.10
D.10	5	5	3	2	4	4	2	2	4	3	4	3	4	5	5	3	3	3	4	4	3.60	4.00	4.00	0.99
E.1	4	4	5	2	4	3	2	2	4	3	3	2	5	3	5	1	5	5	5	3	3.50	3.50	5.00	1.28
E.2	4	5	5	3	3	4	2	3	3	4	3	2	5	4	5	1	5	5	5	5	3.80	4.00	5.00	1.24
E.3	4	4	5	3	4	3	2	2	5	3	5	3	5	4	5	1	5	5	4	5	3.85	4.00	5.00	1.23
F.1	4	2	4	1	4	2	4	2	2	1	4	4	5	3	4	3	1	5	5	5	3.25	4.00	4.00	1.41
F.2	5	4	4	4	4	4	5	2	4	1	4	4	5	3	4	4	4	4	4	2	3.75	4.00	4.00	1.02
F.3	5	3	4	3	4	3	5	2	5	1	4	4	5	3	4	4	4	4	4	2	3.65	4.00	4.00	1.09

Table G.1: Results from the field study user satisfaction questionnaire (*n*=20)