Adaptive Interfaces for Mobile Preference-Based Searching

Ryan Tebbutt Hill

Supervisor: Prof. Janet Wesson
Department of Computing Sciences

December 2009

Submitted in partial fulfilment of the requirements for the degree of
Magister Commercii in the Faculty of Business and Economic Sciences
at the Nelson Mandela Metropolitan University

ACKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Janet Wesson, for her continuous support and guidance shared throughout my postgraduate studies. I would like to thank her for the many hours she spent reviewing the various versions of this dissertation and associated research publications. Her continuous encouragement, guidance and enthusiasm for the project, kept me inspired and motivated to complete this research.

I would also like to thank Mr Danie Venter, from the NMMU Unit for Statistical Consultation, for his recommendations and assistance with the preparation and analysis of the various evaluations conducted during this research.

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

I would like to thank all the Department of Computing Sciences staff members who assisted with the research including the provision of technical support. Thank you to the staff members and fellow students who assisted with the evaluation of prototypes during development.

I would also like to thank the Nelson Mandela Bay Tourism staff for their collaboration with the field studies conducted. Thank you to the tourists, both local and international, who participated in the field studies. Your input was most appreciated.

Lastly, a special word of gratitude to my family and close friends for their support and understanding during the many hours it took to complete this dissertation.

ABSTRACT

Today's mobile computing devices provide a convenient means to search for points-of-interest (POIs) such as restaurants and accommodation. Mobile Preference-Based Search Tools (PBSTs) allow users to identify POIs such as restaurants or accommodation most suited to their needs and constraints using a mobile device. These devices however, have several design constraints including limited screen space and hardware capabilities. Adaptive User Interfaces (AUIs) have been proposed to address these issues but have not been extensively applied to mobile PBSTs such as mobile tourist guides. In addition, AUIs possess several benefits and advantages over static (traditional) interfaces, which do not take a user's preferences, skill set and experience into account. Little research, however, has been conducted into identifying the potential benefits of AUIs for mobile preference-based searching (PBS).

The aim of this research was to determine the extent to which an AUI could improve the effectiveness and user satisfaction of mobile PBS. A literature study was conducted to determine the benefits and limitations of existing mobile PBSTs and determine how these could be improved. The potential benefits of AUIs for mobile PBSTs and a mobile map-based visualisation system were identified. A suitable model for incorporating an AUI into a mobile PBST was identified. The requirements for a mobile PBST were combined with the potentially adaptable objects of a Mobile Map-based Visualisation (MMV) system to provide adaptation suggestions for POInter, an existing mobile tourist guide.

A field study using POInter was conducted in order to measure the extent to which participants agreed with suggestions provided for adapting the information, interaction and visualisation aspects of the system. These results were used to derive adaptation requirements for A-POInter, an adaptive version of POInter. Using a model-based design approach, an AUI was designed and implemented for A-POInter. An extensive field study was then conducted to evaluate the usability of the adaptations provided by A-POInter.

The quantitative and qualitative data collected from the evaluations allowed the usability of A-POInter to be determined. The results of the field study showed that the participants were highly satisfied with the usability and the usefulness of the adaptations provided by A-POInter. Conclusions and recommendations for future work based on the results of the research were then outlined to conclude the dissertation.

Keywords: Adaptive user interfaces, field study, mobile map-based visualisation, mobile preference-based searching, mobile tourist guide, model-based design, points-of-interest.

TABLE OF CONTENTS

List of Figures

List of Tables

Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
AOI	Area-Of-Interest
AUI	Adaptive User Interface
CS&IS	Computer Science & Information Systems
NMMU	Nelson Mandela Metropolitan University
GPS	Global Positioning System
GUI	Graphical User Interface
HCI	Human-Computer Interaction
IHS	Intelligent Help System
IR	Information Retrieval
ITS	Intelligent Tutoring System
IUI	Intelligent User Interface
LBS	Location Based Services
LVD	List View Display
MDE	Model Driven Engineering
MFU	Most Frequently Used
MMV	Mobile Map-based Visualisation
MPUQ	Mobile Phone Usability Questionnaire
MRU	Most Recently Used
MSDN	Microsoft Developer Network
MVD	Map View Display
NESW	North-East-South-West
NMBT	Nelson Mandela Bay Tourism
PBS	Preference-Based Search/Searching
PBST	Preference-Based Search Tool
PDA	Pocket Digital Assistant
POI	Point-Of-Interest
PSSUQ	Post-Study System Usability Questionnaire
QUIS	Questionnaire for User Interaction Satisfaction
SQL	Standard Query Language
SUMI	Software Usability Measurement Inventory
UAS	User-Adaptive System
UMMC	User Monitoring and Modelling Component
XML	eXtensible Markup Language

Chapter 1: Introduction

1.1 Background

Advanced mobile computing devices are becoming more powerful and available to the public at affordable prices. These mobile devices are ideal for searching for and identifying points-of-interest (POIs) such as restaurants or accommodation. A preference-based search tool (PBST) is an application that "assists users in finding multi-attribute products or services that best satisfy their needs, preferences and constraints" (Burigat, Chittaro and Marco 2005). For example, instead of having to select a specific POI from a list, a user could browse a map in order to identify the most suitable POI according to specified criteria, such as a price range and restaurant type (e.g. Italian, with a moderate price range). Search results can be ranked and filtered according to the extent to which they match user-specified preferences.

A mobile map-based visualisation (MMV) system called "*POInter*" was developed in 2007 as part of an Honours Treatise entitled: "*A Mobile Preference-Based Search Tool for Tourism Decision Support* (Hill 2007). The system enables users to identify POIs most suited to their needs and constraints. Search results are superimposed as icons placed upon a map. A green and red bar to the left of a POI icon indicates the percentage match to the user's search criteria (Figure 1.1). An evaluation of POInter showed that preference-based searching (PBS) could be used as an effective search tool to support mobile tourism (Hill and Wesson 2008). The results of the evaluation revealed that users found the system to be highly satisfying and easy to use, therefore providing a highly effective PBST for tourism decision support.

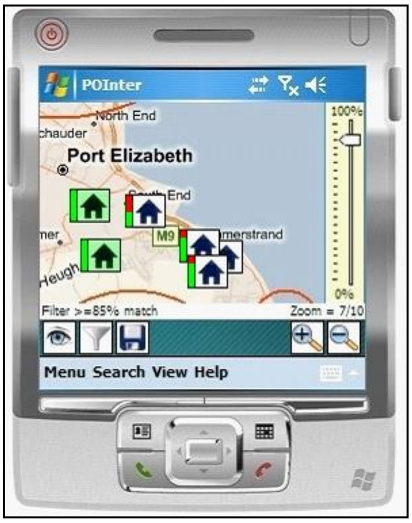


Figure 1.1: POInter visualises POI search results by placing icons on a map (Hill and Wesson 2008)

Despite considerable advances in recent years, mobile devices still fall short of the speed and efficiency possible using a standard PC keyboard and mouse. Mobile interfaces can also be tricky and cumbersome to use when compared to fully-featured desktop graphical user interfaces (GUIs). GUIs are becoming increasingly complex as applications continue to add more features and advanced functionality. Most users will only utilise a small subset of the entire functionality available, even though all possible commands in the system are available all of the time. This persistent availability of all commands significantly decreases the size of the main workspace available to the user, as the surrounding areas become cluttered with interface components such as menus and toolbars. This problem is aggravated when designing mobile applications, as the limited screen size means that optimisation of screen-space usage becomes a key design consideration (Nillson 2009).

As systems continue to grow in terms of complexity and number of features they provide, personalisation has become increasingly important (Findlater and McGrenere 2004; Stuerzlinger, Chapuis, Phillips and Roussel 2006; Preece, Rogers and Sharp 2007). Personalisation is defined as the actions a system takes to provide customised services which match the specific needs of a user, according to information collected whilst interacting with the system. This information can include a user’s interests and preferences which the system stores to create a user profile (Zhang, Karabatis, Chen, Adipat, Dai, Zhang and Wang 2006). Personalisation can be achieved by interpreting and providing assistance based upon the user profile. Personalisation issues include discovering the type of assistance a user requires, learning if and when to interrupt the user and how assistance should be provided (Schiaffino and Amandi 2004). The extent to which a traditional user interface can handle the differences between a user’s preferences, skill set and experience is very limited and does not allow for a high level of personalisation (Alvarez-Cortes, Zayas-Perez, Zarate-Silva and Uresti 2007).

Adaptive User Interfaces (AUIs) have been proposed to address the issues discussed above relating to mobile devices and personalisation. An AUI monitors a user’s activity and attempts to identify usage patterns in order to automatically adjust interface components or content (Stuerzlinger *et al.* 2006; Zhang *et al.* 2006; Alvarez-Cortes *et al.* 2007). The design and implementation of an AUI to support mobile PBS can thus be considered as highly relevant and appropriate.

1.2 Terminology

In order to provide consistent terminology throughout this dissertation, a few general definitions relating to AUIs are discussed below.

Early literature states that an *Intelligent User Interface* (IUI) is the integration of an AUI with an *Intelligent Help System* (IHS) and an *Intelligent Tutoring System* (ITS) (Figure 1.2) (Dieterich, Malinowski, Kühme and Schneider-Hufschmidt 1993; Hefley and Murray 1993). An IHS makes context-sensitive help available and an ITS supports the user in learning to use the system. More recently, IUIs have also been classified as a subfield of Human-Computer Interaction (HCI), which comprises a number of overlapping disciplines including Artificial Intelligence (AI), and Psychology (Figure 1.3). The adaptation process in IUIs is accomplished by utilising AI techniques (Alvarez-Cortes *et al.* 2007). Some authors use the terms dynamic interfaces, IUIs, AUIs, user modelling systems, software/intelligent agents and user-adaptive systems (UAS) interchangeably (Jameson 2002; Alvarez-Cortes *et al.* 2007). For the purposes of this dissertation, the term AUI will be used to indicate an IUI that supports adaptation.

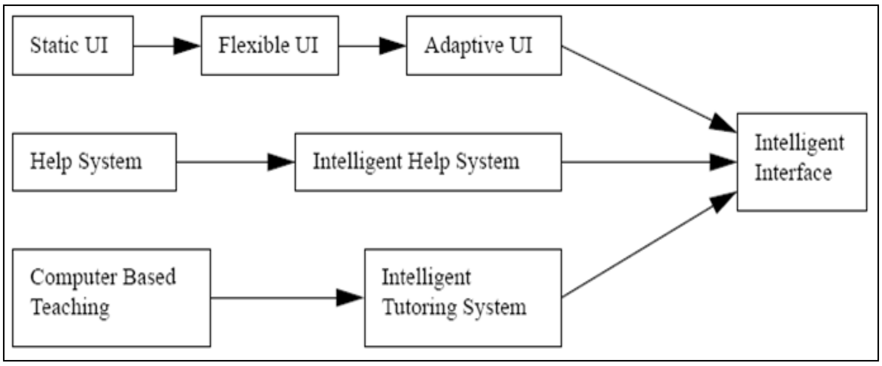


Figure 1.2: Adaptive User Interfaces and Intelligent Interfaces (Dieterich et al. 1993)

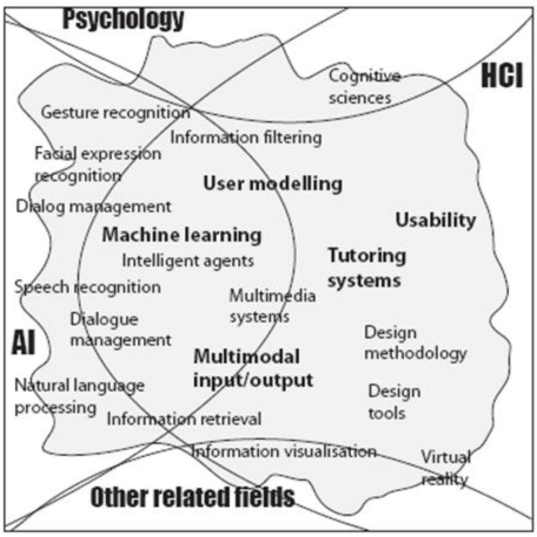


Figure 1.3: IUIs as a multi-disciplinary research area (Alvarez-Cortes et al. 2007)

AUIs can be divided into three classification types: *adaptive*, *adaptable* and *mixed-initiative*. An *adaptive* (or self-adapting) user interface will monitor usage patterns and automatically adjust the interface components or the content the system provides in order to accommodate for different users and behavioural changes. An *adaptable* (or user-adaptable) interface allows manual control of the aforementioned adjustments, usually through specialised help or guidance. Some authors refer to this as a *customisable* interface. With feature-rich software, an adaptable interface is favoured over an adaptive interface, however the latter still has strong support. A reason why adaptable interfaces are preferred is that users may perceive a loss of control when using AUIs (McGrenere, Baecker and Booth 2007). This view is shared by Findlater *et al.* (2004), where users expressed overwhelming support for their personalised menu design. Some users, however, may not recognise the value of interface customisation. For example, test subjects only recognised the performance benefits of placing the most frequently used items near the top of a dynamic menu, after using a static menu for comparison (Findlater and McGrenere 2004).

A *mixed-initiative* approach incorporates both adaptive and adaptable dynamic adaptations. Control over interface changes are shared between the user and the system (Findlater and McGrenere 2004; Alvarez-Cortes *et al.* 2007). This hybrid approach is useful for users that fail to re-customise an adaptable interface when their personal work habits change. The system could then suggest changes to the user based upon what it has determined to be most useful (Gajos, Czerwinski, Tan and Weld 2006). The above approaches can be used to support personalisation in complex software systems (McGrenere *et al.* 2007).

A hierarchical diagram can be used to summarise the different terms discussed in this section and is shown in Figure 1.4 below.

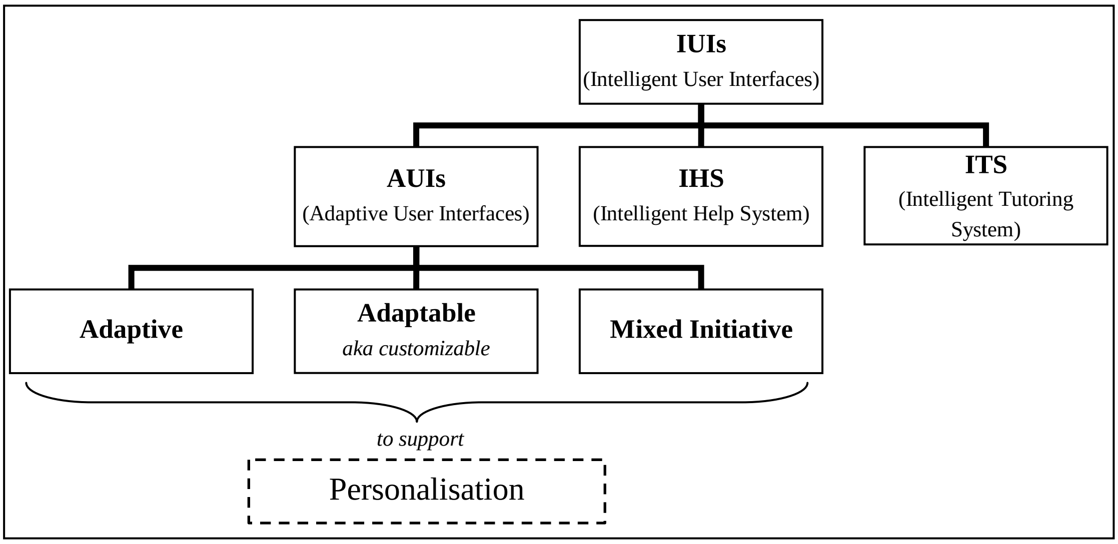


Figure 1.4: Hierarchical representation of dynamic interface terminology

1.3 Model-Based Design

Software implementation is becoming an increasingly complex task, as more advanced functionality and features are required in modern-day applications and systems. Model driven engineering (MDE) aims to reduce the 'gap' between problem and software implementation primarily through the use of models (France 2007). Model-based design allows developers to obtain a better understanding of the system being developed, by providing various abstractions of the system (France 2007; Gomaa and Hussein 2007; Sommerville 2007). An abstraction deliberately leaves out detail by simplifying and picking out the most salient characteristics of a system, forming an important bridge between the analysis and design stages.

The growing complexity of software requirements can overwhelm a developer's ability to effectively maintain mental models of a system during implementation. Depicting models through the use of graphical diagrams can relieve the cognitive burden and accidental complexities associated with maintaining mental models (France 2007). Due to their graphical representation, models are often more understandable than detailed natural language descriptions of system requirements, as they give the developer an overview of the system organisation that can be referred to throughout implementation (Sommerville 2007). In a team development environment, modelling a system provides a mechanism for good communication and allows the system design to be modified in the future with minimal effort. If the source-code accurately and consistently reflects the model design, then the model can be used to debug the code, test the code and contribute towards reusability of the model in future applications (Lapping 2004). The development of a mobile AUI system can also be substantially simplified by following a modelling approach (Nilsson, Floch, Hallsteinsen and Stav 2006). This approach will therefore be used to develop an AUI for POInter.

1.4 Relevance of Research

In recent years, considerable research comparing static interfaces to dynamic interfaces has been conducted (Shneiderman and Maes 1997; Findlater and McGrenere 2004; Sharifi, Deters, Vassileva, Bull and Röbig 2004; Alvarez-Cortes *et al.* 2007). In addition, several mobile AUIs have been developed (Mitrovic, Royo and Mena 2005; Zhang *et al.* 2006). Little research, however, has been conducted into identifying the potential benefits of AUIs for mobile PBS. This research is particularly relevant in a country such as South Africa, where there are over 39-million mobile phone users (nearly 80% of the population) (SouthAfrica.info 2008).

1.5 Thesis Statement

An AUI will provide usability benefits for mobile PBS, specifically in the areas of ease-of-use, usefulness and satisfaction.

1.6 Problem Statement

The aim of this research is to determine the extent to which an AUI can improve the effectiveness and user satisfaction of mobile PBS.

1.7 Research Questions

Several research questions have been identified, namely:

1. What are the limitations of existing mobile PBSTs?
2. What are the potential benefits of AUIs?
3. What are the adaptation requirements for a mobile PBST?
4. How can an AUI be developed to support mobile preference-based searching?
5. What are the benefits of an adaptive mobile PBST?
6. What are the recommendations and conclusions of this research?

1.8 Objectives

The objectives of this project are closely related to the research questions and include the following:

- To investigate the limitations of existing mobile PBSTs;
- To propose requirements for a mobile PBST;
- To identify the potential benefits of AUIs;
- To investigate how AUIs can be used to support mobile PBS;
- To select an appropriate model to support the development of an AUI for a mobile PBST;
- To derive adaptation requirements for a mobile PBST;
- To develop an AUI, called A-POInter, to support mobile PBS;
- To identify the benefits of an adaptive mobile PBST by evaluating the usability of A-POInter and;
- To derive recommendations and conclusions based on the results of this research.

1.9 Scope and Constraints

This research involves developing a prototype incorporating an AUI, based upon the map-based metaphor used for mobile PBS. System adaptations can be classified into four distinct levels or categories: The *information* that is displayed can be adapted to suit the user's current task or context; the *interaction* techniques or interface design can be adapted; the *visualisation* or presentation of the information can be adapted; and the *technology* can be adapted to suit different devices (Reichenbacher 2003).

Technology adaptation is the encoding of information, targeted towards specific (differing) devices, in order to suit their display size, resolution, processing power, etc (Reichenbacher 2003). Device adaptation is a generally low-level implementation related issue that does not depend on user personalisation. This level of adaptation is therefore not relevant to the problem statement and for the purposes of this research, will not be considered.

POInter (Hill and Wesson 2008) was designed specifically for mobile tourism decision support. POInter will be used as a case study for the design and implementation of an AUI. Design of the AUI will focus on user-based adaptation and will therefore not include context-aware adaptations such as location-based services (LBS).

The system will be designed for mobile PDA devices that support Global Positioning System (GPS) and telephonic functionality. POInter was designed for Microsoft Windows Mobile 5.0 PocketPC and the new system will therefore target the latest version of this platform, namely Windows Mobile 6.0 Professional and higher.

This research will focus on identifying the usability benefits of an AUI for a mobile PBST, specifically in the areas of ease-of-use, satisfaction and usefulness. The evaluation will include the measurement of other usability aspects such as control, noticeability and intention-to-use.

1.10 Research Methodology

As this research involves multi-disciplinary aspects, several methodologies will be used throughout the project lifecycle.

A literature study including an analytical review of extant systems will be conducted in Chapters 2-3 in order to answer the first two research questions (Section 1.7). Existing mobile PBSTs and related systems will be examined in order to identify their limitations, determine how these can be improved and propose requirements for a mobile PBST. The use of AUIs with specific regard to mobile applications will be investigated and the limitations of extant systems will be analysed. The proposed requirements for a mobile PBST combined with the identification of adaptable objects in a MMV system will be used to provide adaptation suggestions for POInter (Hill and Wesson 2008), an existing mobile tourist guide.

A field study using POInter will be conducted to measure the extent to which participants agree with the suggested adaptations. The results of this field study will be used to derive adaptation requirements for A-POInter, an adaptive version of POInter.

A model-based design approach will be utilised to complete the design and implementation of A-POInter. Informal user testing will be used to determine the desired type of adaptation and mobile interaction techniques, as well as obtain constructive feedback and suggestions for improvements.

The adaptive system will be subjected to an extensive formal evaluation process. Field testing over an extended period of time will be conducted to determine the benefits of the adaptive system in its actual context of use. Quantitative and qualitative feedback will be collected through the use of a user satisfaction post-test questionnaire.

Critical analysis of the quantitative and qualitative data collected from the evaluations, combined with the literature review, will be used to identify the potential benefits of an adaptive mobile PBST.

Table 1.1 summarises how each research question will be operationalised and the chapter where it will be documented.

Research Question	Methodology	Chapter
1) What are the limitations of existing mobile PBSTs?	Literature Study Extant System Analysis	Chapter 2
2) What are the potential benefits of AUIs?	Literature Study Extant System Analysis	Chapter 3
3) How can an AUI be developed to support mobile PBS?	Requirements Analysis Model-based Design Iterative Development	Chapters 4 - 5
4) What are the benefits of an adaptive mobile PBST?	Field Study	Chapter 6
5) What are the recommendations and conclusions of this research?	Critical Analysis	Chapter 7

Table 1.1: Research methodology

1.11 Structure of Dissertation

Chapter 2 examines existing tools and techniques for mobile searching. Various techniques for searching through large data sets based on explicitly specified user preferences are discussed and their shortcomings outlined. Mobile interaction techniques suited to PBS and MMV systems are analysed. Several visualisation techniques to effectively display PBS results on a mobile device, including displaying off-screen information are described. Requirements for a mobile PBST are also proposed in this chapter.

Various algorithms and techniques to support PBS are discussed in Chapter 2. The algorithm that was used as a basis to develop the multi-criteria PBS algorithm for POInter (Hill and Wesson 2008) is explained and suggestions proposed to further improve the algorithm.

The potential benefits of adaptive interfaces and the different types of adaptation to support mobile PBS are described in Chapter 3. Several issues that static interfaces cannot address such as providing personalisation, taking over tasks, reducing information overflow and providing dynamic help are discussed.

The four main types of adaptation, namely information, interaction, visualisation and technology adaptation, are discussed in Chapter 3. Several components and models to support the different types of adaptation are discussed including a discussion of an existing model for adaptive MMV systems. The potentially adaptable objects of a MMV system are identified. The chapter concludes by selecting and motivating the most appropriate adaptive techniques to support mobile PBS.

The proposed requirements for a mobile PBST are combined with the potentially adaptable objects of a MMV system to propose adaptation suggestions for POInter. Chapter 4 discusses the field study conducted in order to measure the extent to which participants agreed with these adaptation suggestions. The results of a post-test questionnaire containing suggestions for adapting the information, interaction and visualisation aspects of POInter are analysed and findings presented. A set of adaptation requirements for A-POInter are presented in the chapter.

Chapter 5 discusses the design and development of A-POInter. The use of an existing model to support the development of the adaptive system is described. User interface designs for each adaptation requirement identified in Chapter 4 are presented and

discussed. Various aspects and implementation details of the system are discussed, together with the adaptation algorithms used.
Chapter 6 describes the international field study conducted. Extensive user evaluation and testing was performed in order to determine the usability benefits of A-POInter. The chapter also discusses techniques for evaluating AUIs and mobile systems. The suitability of various evaluation instruments and metrics are discussed together with motivation for those selected for the evaluation of A-POInter. The results of this field study are compared to the results of the requirements analysis discussed in Chapter 4.

Chapter 7 concludes the dissertation. The chapter reports on conclusions drawn from the project, summarising what was achieved and highlighting the contributions and benefits of this research. The analysis of the quantitative and qualitative data collected from the field study conducted in Chapter 6, combined with the literature study, are used to determine the benefits of an adaptive mobile PBST. Any problems that were encountered during the research are briefly discussed. The chapter concludes by discussing the contribution of this research as well as providing recommendations for future research and development.

Chapter 2: Mobile Searching Tools and Techniques

2.1 Introduction

This chapter examines existing mobile searching tools and techniques and outlines their shortcomings. A new multi-criteria PBS algorithm that takes these limitations into account was designed by Hill (2007) and implemented in a mobile PBST called POInter. The design and implementation of this algorithm is discussed in Section 2.5.4. Several limitations of the algorithm are discussed and methods to improve the multi-criteria PBS algorithm are proposed. Requirements for a mobile PBST are proposed in Section 2.6.

2.2 Searching Tools and Techniques

Interpreting useful information from large information storage repositories (such as those found on the Internet), can be problematic (Burke, Hammond and Young 1997). Various techniques are available for browsing and sorting large data sets which assist users in identifying search results that best meet their specific needs, preferences and constraints. These techniques include instance-based browsing (Burke *et al.* 1997), standard dynamic query filtering (Dunlop, Ptasinski, Morrison, McCallum, Risbey and Stewart 2004) and preference-based searching (Burigat *et al.* 2005). A discussion of these techniques is given in more detail in the following subsections.

2.2.1 Instance-based Browsing

A technique that bases the retrieval of search results on the analysis of previously retrieved information is known as instance-based browsing. Heuristics are used as in knowledge-based retrieval strategies, to find relevant information. After the system has retrieved POIs matching a user's initial preferences, a user can modify the preferences to yield slightly different results. Instance-based browsing uses domain-level "tweaks" to suggest search results to users. For example, users of the Entrée FindMe system (a restaurant guide for Chicago), can specify options describing their desired restaurant (e.g. Chinese or Seafood) after which the system returns search results matching the user's initial preferences. The user then has access to several "tweaks", such as finding cheaper restaurants or livelier ones (Figure 2.1) (Burke *et al.* 1997). The technique does not support the display and interpretation of search results using a map and is therefore not suited to MMV systems.

Figure 2.1: The Entrée FindMe system showing access to several "tweaks" at the bottom of the screen (Burke et al. 1997)

2.2.2 Standard Dynamic Query Filtering

Dynamic queries provide users with an efficient method to specify queries and visualise the results. Using standard dynamic query filtering, a user specifies his/her preferences via common input controls or "widgets" such as Checkboxes, Radiobuttons and Sliders. POI search results are then filtered according to the specified query criteria and displayed (Dunlop *et al.* 2004). Query results are visualised graphically, usually through the use of maps, scatter-plots or other visualisation techniques. The Taeneb City Guide (Figure 2.2) makes use of standard dynamic query filtering to search for restaurants around the Glasgow city centre using a mobile device. User studies have shown that dynamic queries are more usable and powerful than lists, form filling or natural language systems (Burigat *et al.* 2005).

POIs that do not satisfy all the specified preferences are completely filtered out when using standard dynamic query filtering. A user can only view search results after completing all the search criteria categories in a specific order (Burigat *et al.* 2005).

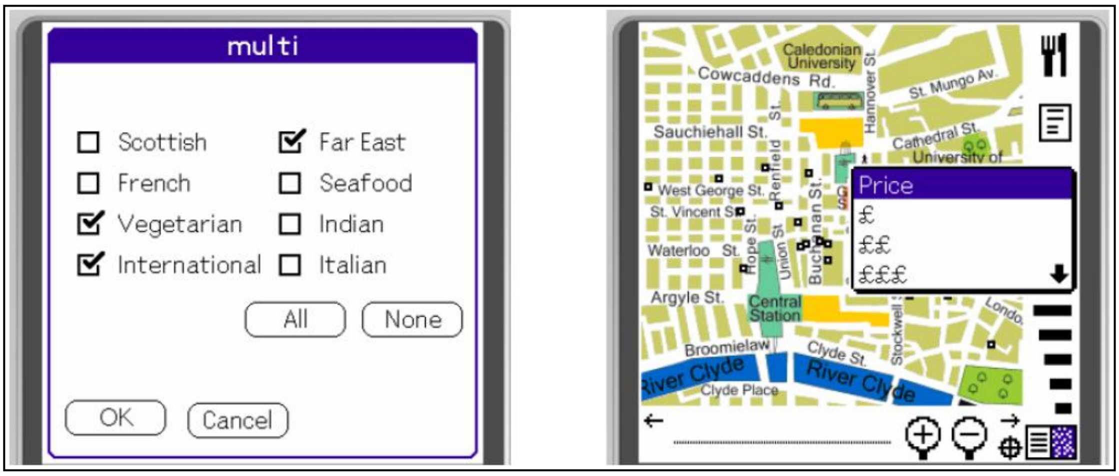


Figure 2.2: Query filters used in the Taeneb City Guide, include restaurant type and a price range (Dunlop et al. 2004)

2.2.3 Preference-based Searching

Preference-based searching (PBS) is an interactive process that identifies the most preferred search result (also known as the target option), based on a set of explicitly stated user preferences (Viappiani, Faltings and Pu 2006). Unlike standard dynamic query filtering, users are not forced to specify all the search criteria before viewing a set of results and can specify search criteria in any order. Additionally, a PBST allows a user to view incremental changes to a set of search criteria as the visualisation of partially satisfied results is supported (Burigat *et al.* 2005). The mobile PBST prototype developed by Burigat *et al.* (2005) uses dynamic querying devices for the controls found in the Taeneb City Guide. Users are, however, able to construct their preferences incrementally and are not forced to specify their criteria in any specific order. The system is able to display partial matches allowing a user to effectively view incremental changes made to their search queries. PBS can therefore be considered as the most effective searching technique for the visualisation and interpretation of MMV information.

Search results in the Burigat *et al.* system (2005) are visualised as icons superimposed on a map. Users can view more than one POI category at once. Different icons are used to represent the category of the POI (e.g. a "knife and fork" represents a restaurant). A coloured vertical bar adjacent to the icon represents the extent to which the POI satisfies the user's query (Figure 2.3). Algorithms and techniques used for PBS are described in more detail in Section 2.5.

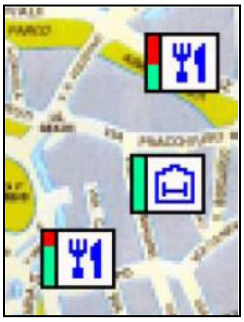


Figure 2.3: POIs are represented by categorical icons overlaid on a map. A coloured vertical bar represents the extent to which a POI matches a user's search criteria (Burigat et al. 2005)

2.3 Interaction Techniques

The PBST system proposed by Burigat *et al.* (2005) implements three different types of query controls to capture user preferences. A rangeslider is used for continuous attributes such as price, a discrete rangeslider is used for ordinal values such as the number of stars of a hotel and standard checkboxes are used for selecting criteria such as services offered by a hotel (Figure 2.4).

A usability evaluation conducted by Burigat *et al.* (2005) revealed that users experienced difficulty manipulating the query controls. Participants felt that the discrete rangeslider should be replaced by standard radioboxes and that the handles of the rangeslider should allow for easier fine-tuning. POInter (Hill and Wesson 2008) addressed these issues by providing simple checkboxes to specify POI search criteria (Figure 2.5). Users select criteria in POInter by ticking checkboxes for classification (list of POI subcategories), price range (ranges from 'R - Inexpensive' to 'RRRRR - Expensive') and star quality (ranges from '★' to '★★★★★').

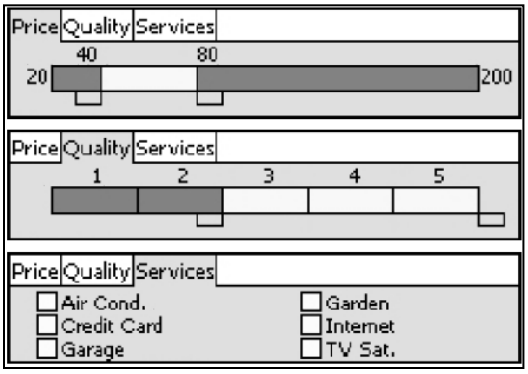


Figure 2.4: The query controls used by Burigat et al. *(2005) include rangesliders (top), Discrete rangesliders (middle) and checkboxes (bottom)*

Figure 2.5: Specifying classification criteria (left) and Star criteria (right) for Accommodation in POInter (Hill 2007)

Panning the map in the Burigat *et al.* system (2005) is achieved by dragging the stylus across the map in the desired direction. The same technique is used in earlier systems such as the Taeneb City Guide (Dunlop *et al.* 2004) as well as more recent systems such as Google Maps Mobile (Google 2009). POInter (Hill and Wesson 2008) used this pen-drag technique but also allowed the user to pan the map using the directional keypad (hardware buttons).

The Taeneb City Guide (Dunlop *et al.* 2004), Burigat *et al.* system (2005) and Google Maps Mobile (Google 2009) all provide the user with onscreen controls to zoom in or out of the map in a stepwise fashion (Figure 2.6). In Google Maps Mobile, a user can also double tap anywhere on the map to zoom into that location.

Figure 2.6: Zooming in Google Maps Mobile (Google 2009) is achieved by step-wise plus or minus icons

In POInter (Hill and Wesson 2008) a user zooms in and out of the map in a stepwise fashion by tapping the 'Plus' or 'Minus' zoom icons located in the bottom-right-hand corner of the toolbar (Figure 2.7) or by selecting a zoom level from the main menu (e.g. *View >> Zoom To >> Level 10 – Street Level*).

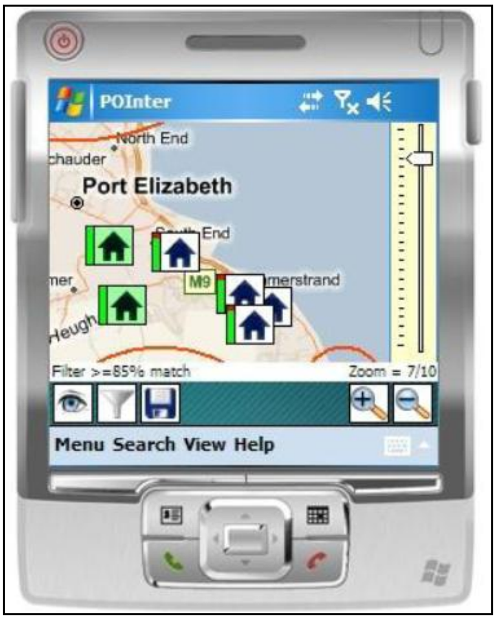


Figure 2.7: POInter supports zooming through the use of step-wise plus or minus icons in the bottom-right-hand corner of the toolbar

Evaluation results revealed that the interaction techniques used in POInter were simple and intuitive and that the users found it easy to enter criteria to search for POIs (Hill and Wesson 2008). A mobile PBST should therefore implement simple controls such as checkboxes for specifying search criteria, feature plus and minus icons for zooming and support pen-drag panning for intuitive manipulation of the map.

2.4 Visualisation Techniques

The most common approach to display search results is using a ranked list (Burigat *et al.* 2005). This is comparable to an index found at the end of a paper-based guidebook. A List View Display (LVD) can be sorted according to various criteria such as alphabetically, categorically or distance to a certain location (Dunlop *et al.* 2004). The Michelin Guide for PDAs allows users to search for hotels and restaurants by entering custom keywords and specifying preferences using basic query controls. The results are displayed using a LVD (Michelin 2006).

A Map View Display (MVD) is an alternative to a LVD, based upon a visualisation technique known as a Starfield display (Dunlop *et al.* 2004). A Starfield display presents large sets of data points on a 2D space (Dunlop and Davidson 2000). The Starfield display technique is ideal for displaying map-based information, as a large data set can be searched efficiently and effectively via the display of spatial information (Dunlop *et al.* 2004). A MVD provides benefits that a LVD cannot, such as visualising the locations of POIs and providing users with a more natural means to interpret different relationships between POIs, such as distance (Figure 2.8). The mobile version of the Zagat Restaurant guide, “Zagat To Go” initially launched for PALM devices in 2000, uses a LVD as the primary means to display search results. A MVD has recently been added to enable searching for nearby restaurants and to visualise driving directions (Zagat 2009).

Figure 2.8: Map View Display (left) vs. List View Display (right) (Dunlop et al. 2004)

POIs that fall outside of the visible portion of the screen should still be visualised (Burigat *et al.* 2005). Off-screen data has traditionally been visualised using an “arrow interface”, by placing arrows at the edge of the screen which point towards the direction in which off-screen POIs can be found. The arrows are usually annotated with the distance to the POI (Figure 2.9).

An alternative to the arrow interface is the Halo Interface which more effectively indicates the location of off-screen POIs (Baudisch and Rosenholtz 2003). A portion of a ring (or “halo”) is drawn at the edge of the screen to indicate an off-screen POI is nearby. The POI is situated at the centre of this ring, which means that as the map is panned, the curvature of the arcs will shrink or grow depending on whether the POI has moved closer or further away (Figure 2.9).

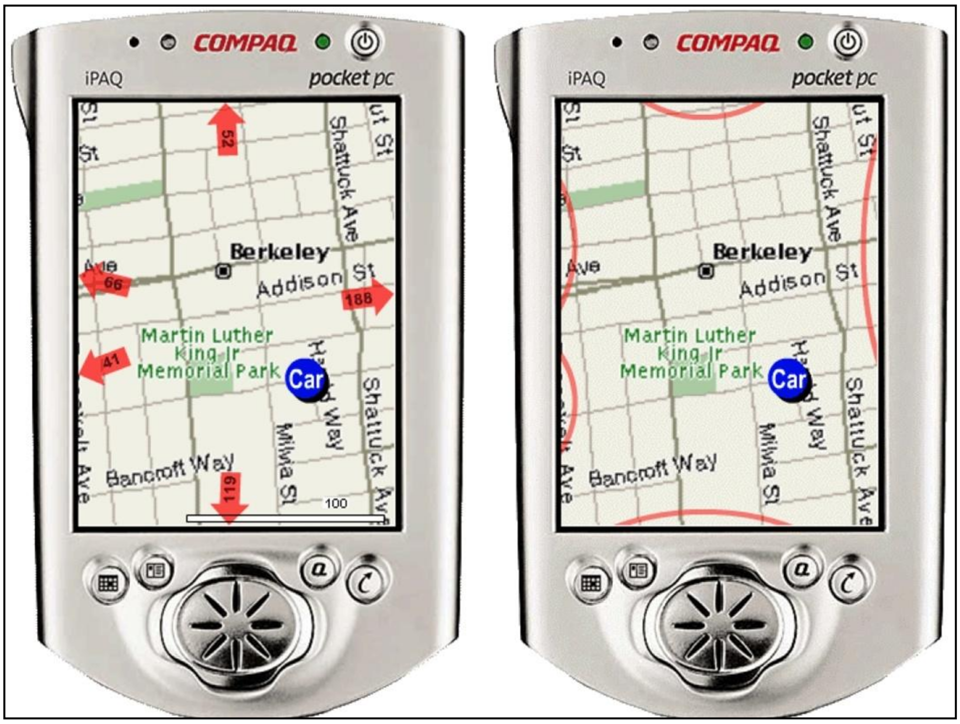


Figure 2.9: The Arrow Interface (left) in comparison with the Halo Interface (right) (Baudisch and Rosenholtz 2003)

A user evaluation of the Halo Interface determined that significant gains in efficiency and satisfaction were obtained when using the Halo interface in comparison to the arrow interface (Baudisch and Rosenholtz 2003). POInter (Hill and Wesson 2008) also utilised the Halo Technique to visualise off-screen POIs (Figure 2.10), with positive ratings for user satisfaction.

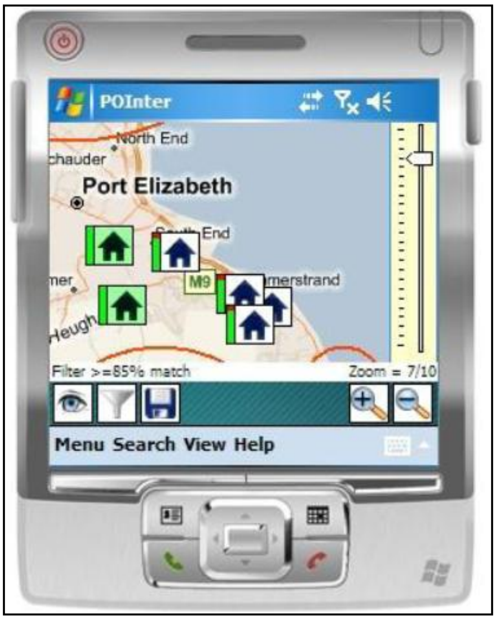


Figure 2.10: The Halo Visualisation Technique was used to visualise off-screen data in POInter (Hill and Wesson 2008)

Additional visualisation issues identified during evaluation of the Burigat *et al.* (2005) system were that a user should be able to hide POIs considered to be irrelevant and that POIs that satisfy all criteria should be made more evident. POInter (Hill and Wesson 2008) met these requirements by providing a filter slider which is used to hide POIs below a specified threshold and highlight POIs that fully satisfy search criteria in green (Figure 2.10).

POI search results in POInter maintain their icon size through all zoom levels (as opposed to conventionally reducing the size of the icon in addition to the map zoom). This can introduce occlusion when multiple POI search results that are close together overlap. This is made even more evident at lower zoom levels (zoomed out). A technique known as "Multiscale Zoom" can be implemented so that the POI icon size decreases as a user zooms out, at a specific scaling rate. This will allow POI icons to be resized, yet remain above the minimum threshold of visibility and readability in zoomed out views (Irani, Gutwin, Partridge and Nezhadasl 2007).

Modern Internet-enabled mobile search tools such as Google Maps Mobile (Google 2009) support different map styles, namely road, satellite or a hybrid of both styles. POInter utilises the road and satellite map styles provided by Microsoft Live Bing Maps (Microsoft 2009). Raster images enable satellite (or aerial photograph) imagery to be downloaded and cached to the device as needed. Satellite imagery enables a user to more easily identify landmarks or real-world objects, whereas simpler road maps allow for a greater contrast between objects that are overlaid on the map such as POIs.

A mobile PBST should therefore include both road and satellite map styles and allow the user to switch between them as desired.

The limitations of existing techniques and tools for mobile search tools are summarised in Table 2.1.

Technique / Tool	Limitation
Searching for Information	
Instance-based Browsing	• Not suited to MMV systems
Standard Dynamic Query Filtering	• Does not support partially satisfied results • Forced to specify all criteria in a specific order before viewing results
Interaction	
Burigat *et al.* (2005) PBST prototype	• Difficult to manipulate query controls
Visualisation	
List View Display (LVD)	• Cannot provide spatial information • Inability to interpret relationships between POIs such as distance
Burigat *et al.* (2005) PBST prototype	• Off-screen POIs are not visualised • Inability to hide irrelevant POIs • POIs satisfying all criteria are not emphasised
POInter (Hill and Wesson 2008)	• Multiple POIs on screen can introduce occlusion

Table 2.1: Summary of limitations of existing mobile tools and techniques

2.5 Preference-based Searching Algorithms

Several techniques and algorithms to support PBS have been proposed and discussed previously (Hill and Wesson 2008). These include Example-Critiquing with Suggestions (Viappiani *et al.* 2006), Preference SQL (Kießling and Köstler 2001) and Real-Valued Dynamic Queries (Fishkin and Stone 1995). These algorithms as well as the PBS algorithm used in POInter are briefly summarised in the subsections that follow.

2.5.1 Example-Critiquing with Suggestions

The Entrée FindMe system discussed in Section 2.2.1 utilises a technique known as example-critiquing, where a candidate set of search results is calculated based on an initial set of explicitly specified preferences. A user critiques the search results given, which revises the preference values in order to return a different set of search results (Burke *et al.* 1997). When this basic example-critiquing technique is used, no history of previous critiques is stored, which means that the entire process could become stuck in a cycle. To avoid this, a preference-model can be added which is used to store previous critiques and suggestions are added to the displayed search results based on these ‘hidden’ preferences (Viappiani *et al.* 2006).

2.5.2 Preference SQL

Search engines typically use the standard structured query language (SQL) to generate search results. SQL is however incapable of understanding ‘soft’ constraints (e.g. “should be” rather than “must be” preferences). An extension to standard SQL called Preference SQL has been developed that allows for soft constraints to be specified using a new keyword, “PREFERRING”. Examples of various preference-based terms that can be used in Preference SQL are given in Table 2.2 below. The Preference SQL examples demonstrate that it is possible to construct complex preferences, whilst adhering to the standard declarative SQL programming style.

Preference SQL Type	Keywords	Example Usage
Approximation	AROUND; BETWEEN	SELECT * FROM trips PREFERRING duration AROUND 14;
Minimisation/Maximisation	LOWEST; HIGHEST	SELECT * FROM apartments PREFERRING HIGHEST(area);
Favourites (positive)	IN	SELECT * FROM programmers PREFERRING exp IN ('java', 'C++');
Dislikes (negative)	<>	SELECT * FROM hotels PREFERRING location <> 'downtown';
Pareto Accumulation (and)	AND	SELECT * FROM computers PREFERRING HIGHEST(main_memory) AND HIGHEST(cpu_speed)
Cascade (ordered importance)	CASCADE	SELECT * FROM computers PREFERRING HIGHEST(main_memory) CASCADE colour IN ('black', 'brown')
Quality Control (restrictions)	BUT ONLY	SELECT * FROM trips PREFERRING start_day AROUND '1999/7/3' AND duration AROUND 14 BUT ONLY DISTANCE(start_day) <=2 AND DISTANCE(duration) <=2

Table 2.2: Preference SQL types and example usage (Kießling and Köstler 2001)

Cosima[T] (Döring 2006) is a prototype website for tourism that utilises Preference SQL. A user is able to specify preferences such as airline carrier, flight class, hotel location and car rental using soft constraints. The system returns ranked search results together with an explanation of how it assigned a specific search result’s ranking.

2.5.3 Real-Valued Dynamic Queries

Standard dynamic query filtering (Section 2.2.2) techniques completely filter out POIs that do not perfectly match a user's specific criteria. In order to visualise partially satisfied results and support PBS, dynamic queries can be enhanced by assigning percentage-based values to search results (Burigat *et al.* 2005). POI search results are assigned a score between 0 (does not meet any search criteria) and 1 (perfect match).

Various techniques for assigning numerical values to search results exist. For example, a piecewise linear function to express that the surface area of an apartment should be at least 30 square metres, but no more than 50 square metres, allowing for a small violation of up to 5 square metres, is shown in Figure 2.11 below. The prototype PBST developed by Burigat *et al.* (2005) utilises real-valued dynamic query filtering to assign scores to POI search results.

$$
\begin{array}{rl}
1 & \textit{if } x < 25 \\
0.2(30 - x) & \textit{if } 25 <= x <= 30 \\
0 & \textit{if } x > 30 \\
0.2(x - 50) & \textit{if } 50 <= x <= 55 \\
1 & \textit{if } x > 55
\end{array}
$$

Figure 2.11: A piecewise linear function representing a set of preferences (Viappiani et al. 2006)

2.5.4 Multi-Criteria PBS Algorithm

2.5.4.1 Overview

A multi-criteria PBS algorithm designed to incorporate the strengths and overcome the shortcomings of existing PBS algorithms was implemented in POInter (Hill and Wesson 2008). Unlike the standard real-value query technique described in Section 2.5.3, the algorithm is able to take into account multiple criteria (e.g. for Accommodation, it can take into account multiple attribute preferences such as the price, star quality, subcategory, etc). The algorithm utilises the "real-valued queries" and linear function techniques (Section 2.5.3), to map numerical values (ranks) onto POI search results.

POInter first builds a candidate result set by selecting rows in a database that satisfy a given SQL query. A total score is calculated for each tuple in the resulting candidate set. This score is divided by a predetermined total weighting to obtain the final percentage ranking. This ranking reflects the extent to which a POI matches the user's preferences (Hill and Wesson 2008). The design of this algorithm is briefly summarised in the following subsections.

2.5.4.2 Building the Candidate Result Set

Standard SQL filter expressions (Section 2.5.2) are used to generate a candidate result set from a given database of tourism POI information. An example SQL query output could therefore be as follows:

```
subcategory = 'Guest House' OR subcategory = 'Hotel' OR price_range =
                    'RR' or price_range = 'RRR'
```

To calculate the predetermined total weighting, the total is initialised to zero. For each check-box selected (i.e. preference) in the 'Subcategory' criteria, a value of 1 is then added to the total. The values added to the weighting for range criteria (such as the price or star rating), must reflect how many ranges the user selected. For example, out of the five price ranges available, if the user selects just one, the value added to the total is 5. If the user selects two price ranges, 4 is added to the running total and so on. The total weighting for the query expressed above is therefore calculated as follows:

```
1 ('Guest House') + 1 ('Hotel') + 4 (price_range 'RR' & 'RRR') = 6
```

2.5.4.3 Calculating Rankings for Search Results

The final rank for a POI search result is calculated by dividing a POI's total score by the total weighting. A total score for each candidate POI is thus initialised to zero. For each check-box ticked in the 'Subcategory' criteria, 1 is added to the total *only if* the candidate POI matches the subcategory ticked. In order to cater for range criteria, the algorithm generates a 1-dimensional array. The optimum value is placed in the cells corresponding to the price ranges selected by the user. The optimum value can be calculated as follows:

```
(total # of cells + 1) – (# of user selections)
```

Figure 2.12 below shows the resulting array if both RR and RRR price ranges were selected by the user.

	Cell 1	*Cell 2*	*Cell 3*	*Cell 4*	*Cell 5*
Price Range:	*(R)*	*(RR)*	*(RRR)*	*(RRRR)*	*(RRRRR)*
Score:	3	**4**	**4**	3	2

Figure 2.12: Price array generated if RR and RRR are selected

The remaining cells receive a score which is calculated as the distance from the closest cell which contains the optimal value. In order to populate the array with numerical values correctly, the array is traversed both forwards and backwards until it has been completely populated.

A value for *price_range* is added to the total score based upon which cell the current candidate falls into. For example, if a candidate accommodation POI was classified as a 'Guest House' and had a price range of RRRR, a resulting total score would be calculated as follows:

```
1 ('Guest House') + 3 (price_range 'RRRR') = 4
```

The candidate POI would therefore receive a final ranking of 67% (The total score 4, divided by the total weighting of 6). After rankings have been assigned to all candidate POIs, search results (including partially satisfied results above 0%) can then be visualised (Hill and Wesson 2008).

2.5.4.4 Searching Algorithm Evaluation

It is important for a PBST to produce search results that are accurate, whilst minimising user effort, in order to improve the effectiveness with which a user can identify target POIs (Viappiani *et al.* 2006). A search engine's effectiveness is dependent on the relevance of items returned by the search algorithm. Evaluating accuracy via user testing could produce biased results, as these results reflect the users' perceived accuracy. Relevancy is also both objective and subjective (Grossman and Frieder 2004; Mihalcea, Mooney, Ghosh and Lee 2008), situational, cognitive and dynamic (Mihalcea *et al.* 2008) therefore making the evaluation of Information Retrieval (IR) systems and search algorithms difficult.

Precision and Recall are two measures commonly used for evaluating the quality of IR results (McNally 2002; Mihalcea *et al.* 2008). Precision and recall can be calculated as per Figure 2.13.

Precision = # of relevant items retrieved / total # of items retrieved.
Recall = # of relevant items retrieved / total # of relevant items

Figure 2.13: Calculating Precision and Recall (Mihalcea et al. 2008)

To achieve a perfect precision score of 1.0, every result retrieved by a search query must be *relevant*. To achieve a perfect recall score of 1.0, a search query must return *all* the relevant items available. A high recall value therefore means that the search algorithm will return a large number of results, but with varying degrees of relevance (McNally 2002; Mihalcea *et al.* 2008).

The multi-criteria PBS algorithm used in POInter makes use of concatenated SQL '`OR`' statements (Section 2.5.4.2) to generate the candidate result set. It can therefore be deduced by inspection that the algorithm always has a perfect recall of 1.0. All results returned by the algorithm are relevant to a certain degree, as they all have a percentage match above zero (Section 2.5.4.3). It can therefore be said that the algorithm has a perfect precision score of 1.0.

A usability study of POInter (Hill 2007) provided insight into the usefulness of the system for tourism decision support and is summarised in Figure 2.14. A high degree of satisfaction was achieved and participants indicated that the information displayed was very useful (mean = 4.82). This correlates with the high ratings received for overall user reactions with POInter (mean = 4.56). It can therefore be concluded that the multi-criteria PBS algorithm was perceived by the users to be accurate.

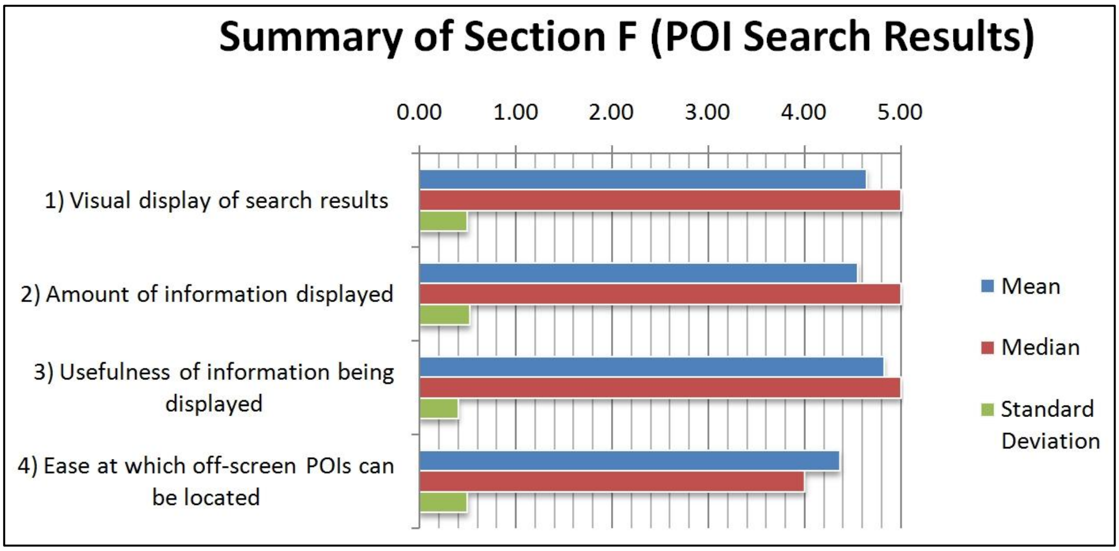


Figure 2.14: Summary of the POI Search Results (using a 5-point Likert scale) in POInter (Hill 2008)

2.5.4.5 Possible Improvements

The multi-criteria PBS algorithm implemented in POInter could be improved. For example, the following extensions could be made:

- Manipulating the weightings used to score criteria, additionally allowing the inclusion of hard constraints; and
- Allowing specification of fine-tuneable criteria.

If a user perceives a price range to hold more importance than a subcategory selection for Accommodation, the weightings can be shifted accordingly to place a higher weighting for price during the calculation of the final score for a POI search result. At the same time, hard constraints can be selected (e.g. the subcategory for Accommodation *must* be a "Guest House" and nothing else).

The multi-criteria PBS algorithm does not support fine-tuned selection for numerical value criteria (e.g. price). Users have to specify price range criteria as "R (inexpensive); RRR (moderately priced); RRRRR (very expensive)" etc. The algorithm could be enhanced by allowing a user to select and fine-tune an exact numerical range (e.g. R150-R200 pp/per night). The criteria score for numerical-based criteria could then be calculated by mapping values using a linear function as discussed in Section 2.5.3.

Requirements for a mobile PBST based on the above discussions are summarised in Section 2.6.

2.6 Requirements for a Mobile Preference-Based Search Tool

The following subsections propose functional and user interface requirements of a mobile PBST based on the investigation into extant systems and techniques discussed in Sections 2.2-2.5.

2.6.1 Functional Requirements

Based on the discussion of tools and techniques for browsing and sorting large datasets discussed in Section 2.2, the following functional requirements for a PBST are proposed:

1. Allow users to explicitly state search criteria.
2. Allow users to view incremental changes to a set of search criteria.
3. Allow users to specify search criteria in any order.
4. Display search results as icons superimposed on a map.
5. Allow the user to view more than one POI category simultaneously.
6. Visualise partially satisfied results.

2.6.2 Interaction Requirements

Based on the interaction techniques for mobile PBSTs discussed in Section 2.3, the following interaction requirements for a PBST are proposed:

1. Use simple checkboxes to specify search criteria.

2. Support panning the map by dragging the stylus across the map in the desired direction.
3. Support zooming the map in a step-wise manner using simple plus and minus icons.

2.6.3 Visualisation Requirements

Based on the visualisation techniques for mobile PBSTs discussed in Section 2.4, the following visualisation requirements for a mobile PBST are proposed:

1. Display search results using a MVD.
2. Visualise off-screen POIs using the Halo technique.
3. Highlight POIs that perfectly match a user's search criteria.
4. Allow the filtering of search results.
5. Prevent occlusion when visualising multiple POI search results.
6. Support both road and satellite map-styles.

2.6.4 Preference-Based Search Algorithms

Based on the PBS algorithms discussed in Section 2.5, the following requirements for a mobile PBS search algorithm are proposed:

1. Support searching using multiple attribute preferences (e.g. Price, Star quality, subcategory, etc).
2. Utilise "real-valued queries" and linear function techniques to rank POI search results.
3. Produce accurate search results by returning only relevant results (high precision) and all the relevant items available (high recall).
4. Allow the manipulation of the weightings (importance) of certain search criteria.
5. Allow both soft and hard constraints to be specified.
6. Allow specific fine-tuning of range-based search criteria.

2.7 Conclusion

Several limitations of existing mobile PBSTs and techniques were identified and summarised in Table 2.1. POInter addressed several of these limitations, but some limitations still exist. For example the "Multiscale Zoom" technique was identified as one possible solution to overcome an occlusion issue when displaying many search results (Section 2.4).

Several shortcomings of existing PBS algorithms were identified in Section 2.5. These include implicit use of preference models and the difficulty in expressing soft constraints. Existing algorithms were identified as not supporting searching using multiple criteria and attribute preferences. A multi-criteria PBS algorithm was developed by Hill (2008) to address these limitations. Positive features such as real-valued queries and linear function techniques were incorporated into the new multi-criteria PBS algorithm. The design of this algorithm was summarised in Section 2.5.4. It was motivated that the algorithm has perfect precision and relevance. Usability testing of POInter revealed that the algorithm was perceived by users to be accurate.

Several suggestions to further improve the multi-criteria PBS algorithm were identified in Section 2.5. These include supporting fine-tuneable criteria and dynamic weightings in order to place more importance on certain attributes and allow hard-constraints to be specified. For example, that the ‘Accommodation’ subcategory type *must* be met.

Requirements for a mobile PBST were proposed in Section 2.6 based on the review of existing mobile PBSTs. These requirements were classified under functional requirements, interaction requirements, visualisation requirements and PBS algorithm requirements. These requirements can be used to design a mobile PBST that supports effective browsing and searching for POIs.

The following chapter discusses adaptive interfaces and examines adaptation types, models and algorithms in order to select the most appropriate adaptive techniques to support mobile PBS.

Chapter 3: Adaptive User Interfaces

3.1 Introduction

This chapter investigates the potential benefits of Adaptive User Interfaces (AUIs) for mobile PBS. Various types of adaptation are discussed and those most suited to mobile PBS are identified. Several models and components to support the adaptation techniques are examined together with a discussion of the Proteus Model, a model selected that supports the adaptation of an MMV system. [1]

3.2 Benefits of Adaptive User Interfaces

Several authors have demonstrated that adaptive interfaces are preferred to static interfaces (Alvarez-Cortes *et al.* 2007). Adapting GUIs for mobile computing is a difficult issue as the design is constrained to the capabilities of the mobile device (Mitrovic *et al.* 2005).

AUIs attempt to solve problems that current static interfaces cannot address. These include creating personalised systems (functionality), taking over tasks from the user (task allocation or partitioning), reducing information overflow (interface transformation) and providing help on using new and complex applications (user adaptation) (Meyer, Yakemovic and Harris 1993; Alvarez-Cortes *et al.* 2007). These four aspects are described in more detail in the subsections that follow.

3.2.1 Functionality

Users have differing behavioural patterns, habits, preferences and work methodologies. An AUI should take these differences into account and provide a personalised method for interaction with the system (Alvarez-Cortes *et al.* 2007). AUIs are beneficial when there are more interface options available than the screen can accommodate simultaneously or if a user only makes use of a subset of the available options (Billsus, Brunk, Evans, Gladish and Pazzani 2002). In this way, the system can adapt the functionality available to suit individual users (Meyer *et al.* 1993).

[1] Portions of this chapter were published in the proceedings of SATNAC 2009 (Hill and Wesson 2009).

Many commercial applications have implemented simple AUIs by allowing the user to customise the interface in predefined ways, such as by adding or removing menu or toolbar functions or moving the position of toolbars. A well known example of this is seen in Microsoft's Office 2000 suite of applications (McGrenere *et al.* 2007). When a menu in an Office application is opened, a "short" menu, which contains only a subset of the complete menu available, is shown. A user can access the full menu by clicking on or hovering over the double down-arrow at the bottom of the short menu (Figure 3.1).

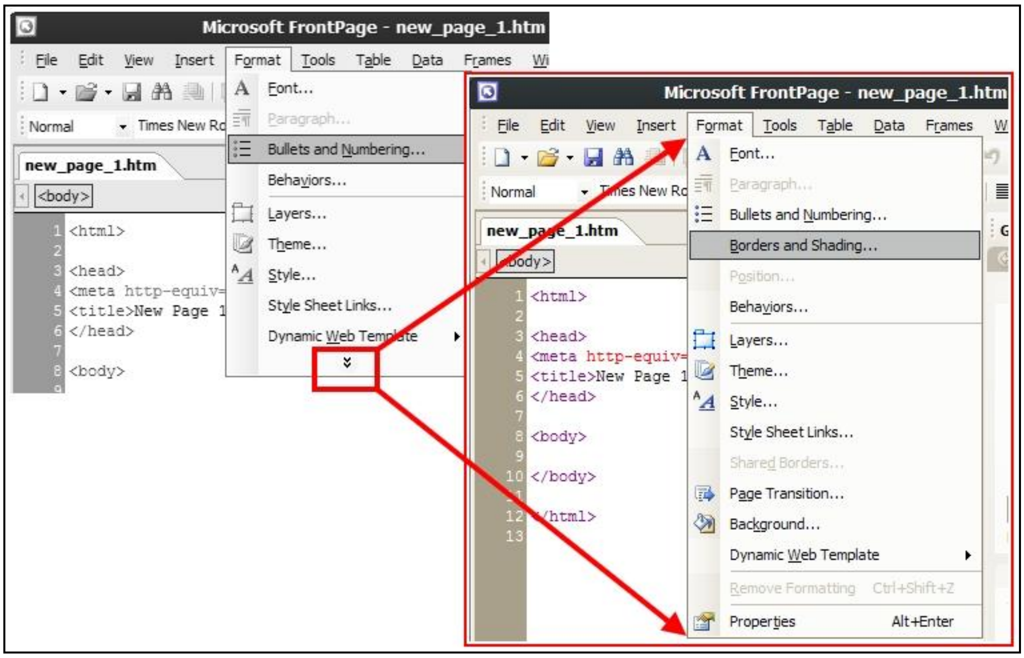


Figure 3.1: Accessing a "full" menu from a customised "short" menu in Microsoft Frontpage 2003

If a user selects an item from the full menu, it will appear in the short menu from that point onwards. Similarly, after a period of non-use, items may be removed from the short menu. Users can only turn the adaptive menu system on or off and reset the data collected. They are unable to view or change the underlying model maintained by the system (McGrenere *et al.* 2007). The Start Menu in Microsoft Windows XP/Vista features an adaptive portion, whereby program shortcuts are generated automatically and sorted based upon frequency of use. Users can specify the number of recent programs (0-30) to display in the list (Gajos *et al.* 2006). Usability studies have shown that adaptive menus have a larger positive benefit on small screen displays than larger desktop applications. Users are also more likely to use the adaptive menu suggestions in a small screen application than in a desktop application (Findlater and McGrenere 2008).

Explicit data capturing techniques can be used to obtain user preferences to create an initial user profile (Alvarez-Cortes *et al.* 2007). For example, a user's age, gender, location, purchase history, content and layout preferences can be recorded as a basis to recognise patterns in user preferences.

3.2.2 Task Allocation or Partitioning

An AUI can look at the current task, understand it, recognise the user's intent and automatically take over the task completely or partially, allowing the user to focus on other more important activities (Meyer *et al.* 1993; Alvarez-Cortes *et al.* 2007). Implicit data gathering techniques can be used, to discover patterns in data logs. Data can also be gathered explicitly by capturing input from the user to form a user profile (Alvarez-Cortes *et al.* 2007).

Examples of implicit data gathering techniques include the analysis of a navigation history or cookies in a web browser. If a user regularly visits a particular website or views particular types of items, a pattern can be identified and the interface can be altered accordingly to display modified content accordingly. The discovery of these patterns allows content to be filtered and delivered based on assumptions made regarding the user's behavioural history (Alvarez-Cortes *et al.* 2007). Recommender systems used in e-commerce websites such as Amazon.com commonly make use of both implicit and explicit data capturing to build a user profile in order to present items that are of likely interest to the user (Amazon 2008). The recommender system used by Amazon.com (Table 3.1) combines personalised recommendations (items based on the user's past behaviour), social recommendations (items based on similar users) and item recommendations (items based on similar items) (MacManus 2009). Implicitly captured data is used by the recommender system to streamline and channel preferred content to the user, while explicitly captured data (such as contact and payment details) is used to speed up and simplify the checkout process for returning customers (Amazon 2008).

User's browsing history	User's purchase history
Actual items	New releases (item recommendation)
Related items (item recommendation)	Related items (item recommendation)
What others purchased (social recommendation)	What others purchased (social recommendation)

Table 3.1: The Amazon.com personalised recommender system uses a combination of personalised, social and item recommendations (MacManus 2009)

3.2.3 Interface Transformation

The information overflow associated with finding information in complex systems or large databases can be reduced through the use of AUIs. Irrelevant information can be filtered out, thereby reducing the user's cognitive load. Content filtering and delivery are enabled via assumptions made about user behaviour after analysing patterns discovered through implicit data gathering (Alvarez-Cortes *et al.* 2007). Tasks can be made easier by modifying the communication style, content and form of information that is displayed (Meyer *et al.* 1993).

Consider the following example: A personalised restaurant finder locates nearby restaurants that match a user's interests. Using adaptive personalisation, the system could automatically learn a user's preferences for restaurants. A specific restaurant could be marked as favourable by positive actions taken by the user, such as contacting the POI or requesting driving directions. The presentation or visualisation of POI search results could then be customised. If the search results are presented in a LVD, an adaptive interface could sort the list not only by distance, but by placing the most personally useful options at the top of the list (based on location as well as personalisation) (Billsus *et al.* 2002). Care must be taken to select the correct data to visualise when using a MVD. Visualising insufficient data might lead a user into selecting a suboptimal POI. Similarly, visualising unnecessary data will make it difficult to interpret the search results. Users should be able to quickly and accurately identify optimal POIs (Chittaro 2006).

Approaches on how to improve the presentation of information on mobile devices can be classified into two categories, namely content summarisation and information visualisation. Instead of displaying the entire content to a user (for example a set of search results), only a summary is presented to the user. The user can then determine whether the summary is interesting and whether he/she would like to see the entire content (detailed search results). Unnecessary browsing of irrelevant information is thus avoided. The *Focus+Context* visualisation technique enables users to focus on important parts of the content (usually displayed at full size and detail) whilst keeping the surrounding context in peripheral view (usually displayed at a reduced size or simplified way) (Huot and Lecolinet 2007). Several Focus+Context techniques such as fisheye techniques and semantic zooming are also applicable to mobile devices (Zhang *et al.* 2006).

A fisheye interface visualises focus information at a higher magnification level than context information (Hornbæk and Hertzum 2007). A fisheye menu visualises a large list by showing a section of information in focus whilst gradually decreasing the font size of surrounding menu items to provide context. A fisheye menu can utilise several different designs (Figure 3.2), each of which impact on usability either positively or negatively depending on factors such as list size or hierarchical menu structure. An overview menu cannot show all menu items simultaneously as it does not make use of any distortion. The menu items in focus reflect the position of the mouse cursor. For example, the menu scrolls upwards when the mouse is moved downwards. With a multifocus menu, important menu items which fall outside of the current focus area have a larger font size, eliminating the need for index letters (Hornbæk and Hertzum 2007).

Figure 3.2: A fisheye (left), overview (middle) and multi-focus (right) menu (Hornbæk and Hertzum 2007)

A static split-menu moves or copies menu items to a top partition. An adaptive split-menu dynamically controls which items appear in the top partition based upon frequency and recent usage. An adaptable split-menu (Figure 3.3) allows the user to position menu items.

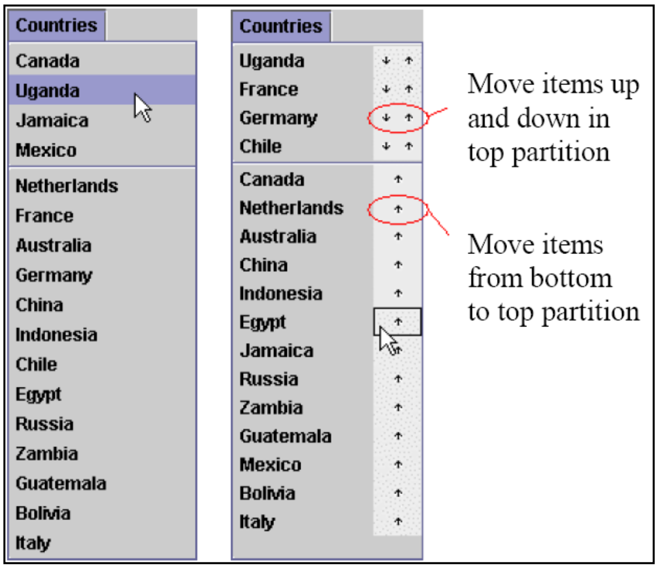


Figure 3.3: A Static Split-Menu (left) and an Adaptable Split-Menu (right) (Findlater and McGrenere 2004)

Geometric zooming simply views a section of a visualisation at a higher magnification level. In contrast, semantic zooming changes the information content to be visualised by adding or removing detail (Nestor, O'Malley, Healy, Quigley and Thiel 2007). For example, as a user zooms in on a POI represented as a dot on a map, it may become a tiny square box or an icon, then a box with a one word label, then contain a longer label, before becoming a rectangle filled with text and a picture (University of Michigan Digital Library Project 1996). POI details that could be revealed when zooming in could include what type of food is served there, its specific price range or the POI name. This is similar to the Multiscale Zoom technique (Irani *et al.* 2007) mentioned in Section 2.4, that can prevent occlusion of multiple POI search results.

The “SpiralList” (Huot and Lecolinet 2006) is a Focus+Context visualisation technique for interacting with large lists on mobile devices. A SpiralList improves the overall visibility of items by providing a global view of the entire list. The spiralled list makes it possible to show more items simultaneously than would be possible with a linear list (Figure 3.4). Labels appearing on a subsection of the outer revolution of the spiral (located at the bottom-right of Figure 3.4) form the Focus area and are fully visible. Remaining labels spiral inwards and are truncated to form the Context area. The first three letters are shown on the outer revolution, followed by 2 letters, etc (Huot and Lecolinet 2006).

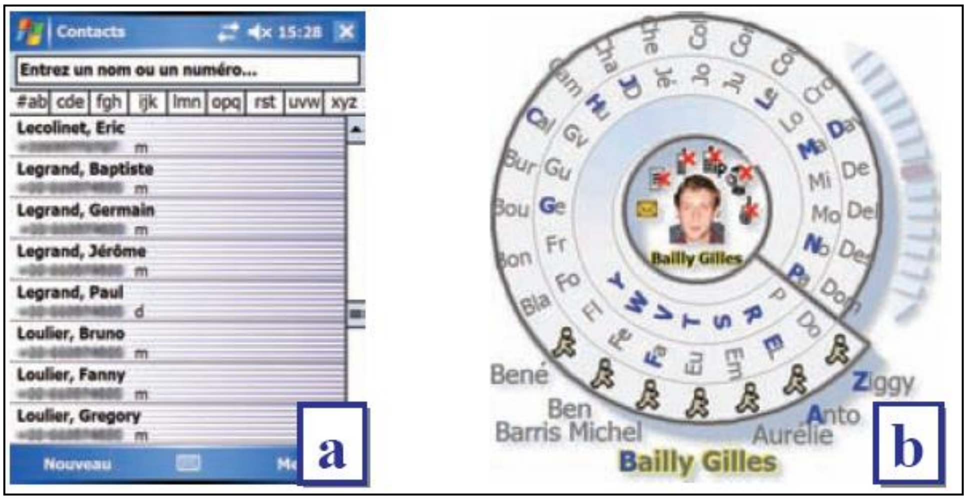


Figure 3.4: The Windows Mobile 5.0 address book display (left) vs. a SpiralList (right) displaying 100 contacts (Huot and Lecolinet 2006)

A preliminary pilot study showed that the SpiralList was well received and evaluations were considered to be time efficient. For lists exceeding 100 items however, actual time measurements gave disappointing results when compared to traditional alphabetised lists. Consequently, it was established that while the SpiralList does provide an efficient solution for displaying large lists on mobile devices, it is most suited for lists that do not exceed 100 items (Huot and Lecolinet 2007).

A new technique termed “SnailList” (Huot and Lecolinet 2007) is an improvement over the SpiralList, which attempts to improve the efficiency when interacting with lists containing over 100 items. The context zone contains all the letters in the alphabet (Figure 3.5). The level of detail for all items is the same (only one character). The user selects a letter to make the focus appear on the selected item. All items starting with the selected letter then appear on the spiral as a new sub-list, which is concatenated to the end of the current context zone. Users continue to “drill-down” until locating the desired item. Additionally, item labels are no longer displayed with text rotation (a feature of the SpiralList determined to negatively affect efficiency) (Huot and Lecolinet 2007).

A usability study showed that the SnailList significantly reduced the error rate (approximately 3.7 times lower) without any loss of performance when interacting with the thumb to manipulate large lists on a mobile device (Huot and Lecolinet 2007). For mobile search tools that utilise a LVD and produce a large number of search results (over 100), the SnailList is therefore the most efficient technique that is able to provide a global view of the entire list.

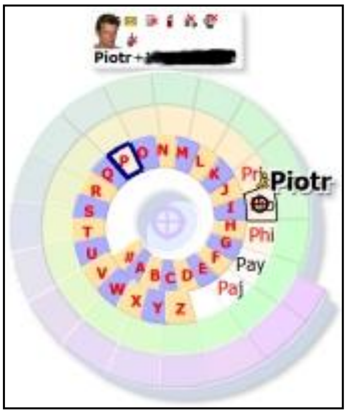


Figure 3.5: The SnailList is an improvement over the SpiralList (Huot and Lecolinet 2007)

3.2.4 User Adaptation

Some software might be too complicated to operate for first-time users without a certain degree of help. AUIs can detect and correct user misconceptions, explain certain functions, provide intelligent tutoring and provide information that would simplify certain tasks (Meyer *et al.* 1993; Alvarez-Cortes *et al.* 2007). Additionally, the first use should provide an acceptable, non-personalised experience. The benefits of the AUI should then become apparent within the first few uses. The transition from a non-personalised to a personalised interface should be a smooth one (Billsus *et al.* 2002).

The focus of this research will be on creating a personalised system and reducing information overflow, as these aspects are applicable to mobile PBS using AUIs and are therefore most relevant to this research.

3.3 Types of Adaptation

Several authors have categorised system adaptations into four distinct types or levels (Reichenbacher 2003; Cena, Console, Gena, Goy, Levi, Modeo and Torre 2006; Zhang *et al.* 2006):

1. Information (Data / Content) Adaptation
2. Interaction (User Interface) Adaptation
3. Visualisation (Presentation) Adaptation
4. Technology (Device) Adaptation

Reichenbacher (2003) listed and grouped adaptable objects for a mobile cartographic system into the first three adaptation types listed above. For example, the content of geographical *information* can be adapted to user, activities and context, the *interaction* style can be adapted by adjusting availability or granularity of controls and the *visualisation* method used can be the object of adaptation. Potentially adaptable objects for mobile cartographic systems as proposed by Reichenbacher (2004) are summarised in Table 3.2.

Information	Interaction	Visualisation
Encoding **Amount** **Level of Detail** **Map Features** – Selection – Classification – Grouping – Geographic Area of Interest	**Function** – Availability (available, hidden, disabled) – Granularity **Map Interaction** – Modality (visual, acoustic, cross-mode) – Style (point and click, forms, menus, queries, natural language – Mode (select, pan, zoom, enter) **Map Functions** – Pan – Zoom – Select Map Area/Layer/Object – Point to – Show Attributes – Calculate Distance/Perimeter/ Area – Redraw	**Map Layout** – Title – Legend – Scale Bar **Map Style** – Level of Detail – Generalisation – Orientation – Scale – Section (spatial focus) – Method/form/graphic structure **Dimension** (2D, 3D/perspective) **Graphical Elements** – Colour – Value – Size – Form – Orientation – Pattern – Clarity – Position – Dimension – Opacity **Typographical Elements** – Font – Style – Symbolisation

Table 3.2: Potentially adaptable objects in a mobile cartographic system (Summarised from Reichenbacher 2004)

These three main types of adaptation listed are relevant to this research and a discussion with examples relating directly to a mobile PBST for tourism decision support are described in more detail in the sections below.

3.3.1 Information Adaptation

The information (data or content) in a system that is presented to the user can be adapted to suit the user's activities and context of use (Reichenbacher 2003). For example, a summary of information (e.g. search results) can be presented to the user, after which the user can decide if it is relevant and if he/she would like to examine the content in more detail (Zhang *et al.* 2006).

Information adaptable objects from Table 3.2 applicable to a mobile PBST could include remembering and suggesting preferences such as search criteria or categories of POIs, selecting a default area-of-interest (AOI), adjusting the level of detail (i.e. what information to overlay on the map) and the filtering of search results (affecting the amount of information displayed).

3.3.2 Interaction Adaptation

Based upon the user's activities, context of use and explicit customisation, the system can adapt the user interface or interaction (Reichenbacher 2003). One user may prefer to zoom in or out of a map by clicking on plus or minus icons, whilst another user might prefer to zoom by drawing a rectangular bounding box around an AOI.

Interaction adaptable objects from Table 3.2 applicable to a mobile PBST include adapting the mode and style used to pan and zoom into/out of a map, techniques used to interact with off-screen content and adjusting the functionality, availability and granularity of menu structures.

3.3.3 Visualisation Adaptation

Visualisation adaptation concerns how information will be presented to the user. For example, a map's scale or zoom level could be automatically adjusted and the amount of POIs filtered based upon how fast the user is travelling (Reichenbacher 2003). Many techniques for adaptive visualisation such as the Focus + Context (Huot and Lecolinet 2007) and Fish-Eye views (Hornbæk and Hertzum 2007) exist, as discussed in Section 3.1.3.

Additional adaptable functionality from Table 3.2 applicable to a mobile PBST could include whether the user prefers portrait or landscape mode and the customisation of the symbols used for all graphical elements on the screen, especially those overlaid on the map. The Multiscale Zoom or semantic zooming techniques discussed in Section

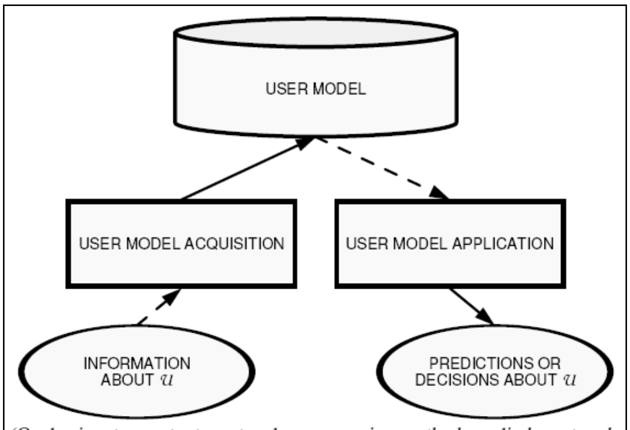
3.2.3 can be used to dynamically alter POI icon sizes or accompanying levels of detail according to zoom level. Customisation of elements could include the opacity, values, colours used (e.g. night mode), size, form, orientation, font, style, etc.

3.4 Model-Based Design

The benefits of model-based design with specific reference to the development of AUIs were briefly discussed in Chapter 1. The following section further motivates the importance of using a modelling approach during the development of an AUI, through the investigation of applicable models for an adaptive mobile map-based visualisation (MMV) system.

3.4.1 Models for Adaptive User Interfaces

In order for a system to provide adaptation, it needs to learn something about each individual user. Jameson (2002) proposed a general schema that can be applied to any AUI system (Figure 3.6). An AUI records information about the current user (such as the previous actions a user has performed, or choices he/she made). During the user model acquisition process, the system performs learning or inferences based upon this information in order to build a user model. The system then applies the user model to the relevant features based on the current context, in order to provide adaptation.



(Ovals: input or output; rectangles: processing methods; cylinder: stored information. Dotted arrows: use of information; solid arrows: production of results.)

Figure 3.6: General schema for processing in an AUI (Jameson 2002)

The process of adaptation takes several steps. Adaptation is triggered by substantial changes between various states of user, activity, information demand or technology. These differences or change measures have to be defined in order to establish threshold values which will control when adaptation is triggered.

Reichenbacher (2003) proposed a conceptual framework to support adaptive mobile cartography (Figure 3.7). The adaptation target is the context or elements to which geo-information visualisation is adapted. These elements are stored in the user profile and are either recognised by the system (implicitly) or predefined (explicitly) by the user. A suitable adaptation strategy must then be selected and the adaptation process modelled. The modelling process involves selecting the elements to adapt (adaptee) and the methods to perform the adaptation (the adapter). The adaptation can then be executed (Reichenbacher 2003).

A middleware architecture and run-time model has been proposed by Nilsson *et al.* (2006) that closely follows Reichenbacher's framework. The middleware has three main functions. It must detect context changes, reason about these changes in order to make decisions about what adaptation to perform and lastly, must perform the adaptation (Figure 3.8).

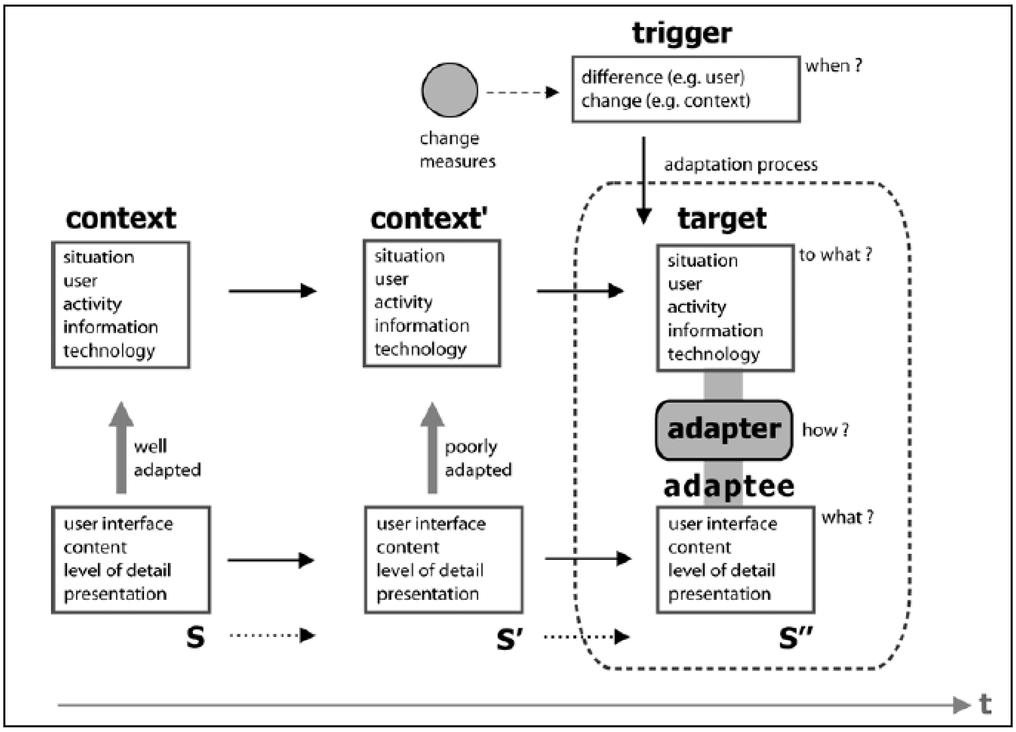


Figure 3.7: Conceptual framework of mobile cartography and adaptation components (Reichenbacher 2003)

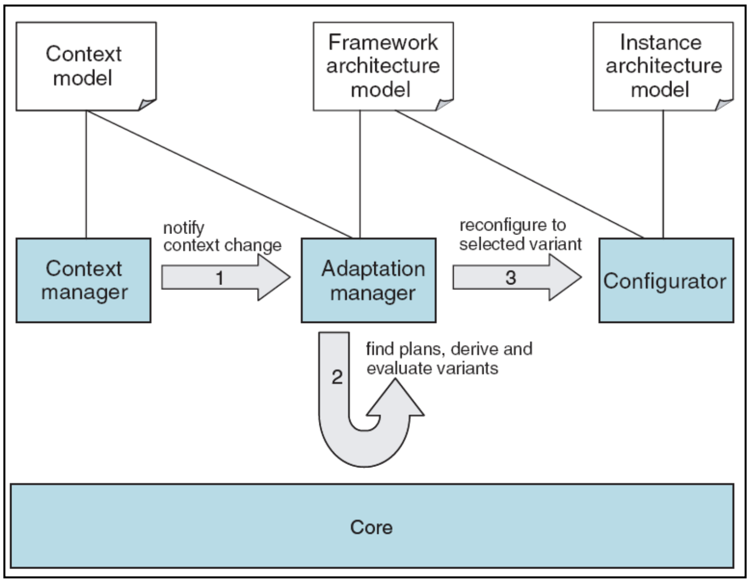


Figure 3.8: Middleware architecture and run-time model to support adaptation (Nilsson et al. 2006)

The simple model proposed by Alvarez-Cortez *et al.* (2007) shown in Figure 3.9 corresponds to the models proposed by Reichenbacher (2003) and Nilsson *et al.* (2006). A User Model collects, processes and displays customised information based on context. The User Model collects information from the user both implicitly and explicitly (Alvarez-Cortes *et al.* 2007).

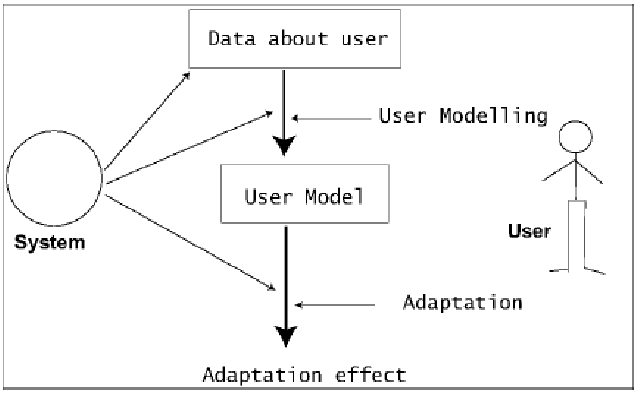


Figure 3.9: System and user perspectives for a user model (Alvarez-Cortes et al. 2007)

The Proteus Model (van Tonder and Wesson 2008) is a new model designed to support a wide range of adaptations and aims to facilitate the development of adaptive MMV systems. The Proteus Model was designed to address the shortcomings of the UbiquiTO architecture (Cena *et al.* 2006) and the Mobile Cartographic Framework (Reichenbacher 2003) by incorporating additional components such as an Adaptation Engine (Figure 3.10) to facilitate information, interaction and visualisation adaptation in response to user behaviour, tasks and context and was therefore considered appropriate for this research.

A prototype MMV system called MediaMaps was developed as proof of concept of the Proteus Model (van Tonder and Wesson 2008), but the model has not yet been used to incorporate adaptation into an existing MMV system. The Proteus Model supports adaptation in the three main areas, namely information, interaction and visualisation adaptation. The Proteus Model is described in more detail in the next section.

3.4.2 The Proteus Model

The Proteus Model (Figure 3.10) incorporates four main groups of components to facilitate adaptation in MMV systems, namely the Data Model, Knowledge Base, Adaptation Engine and User Monitoring and Modelling Component (UMMC).

3.4.2.1 Data Model

The Data Model contains the main data of the system which is separated into map data (image tiles) and additional data, such as POIs that can be overlaid or visualised in the context of the current map shown (van Tonder and Wesson 2008).

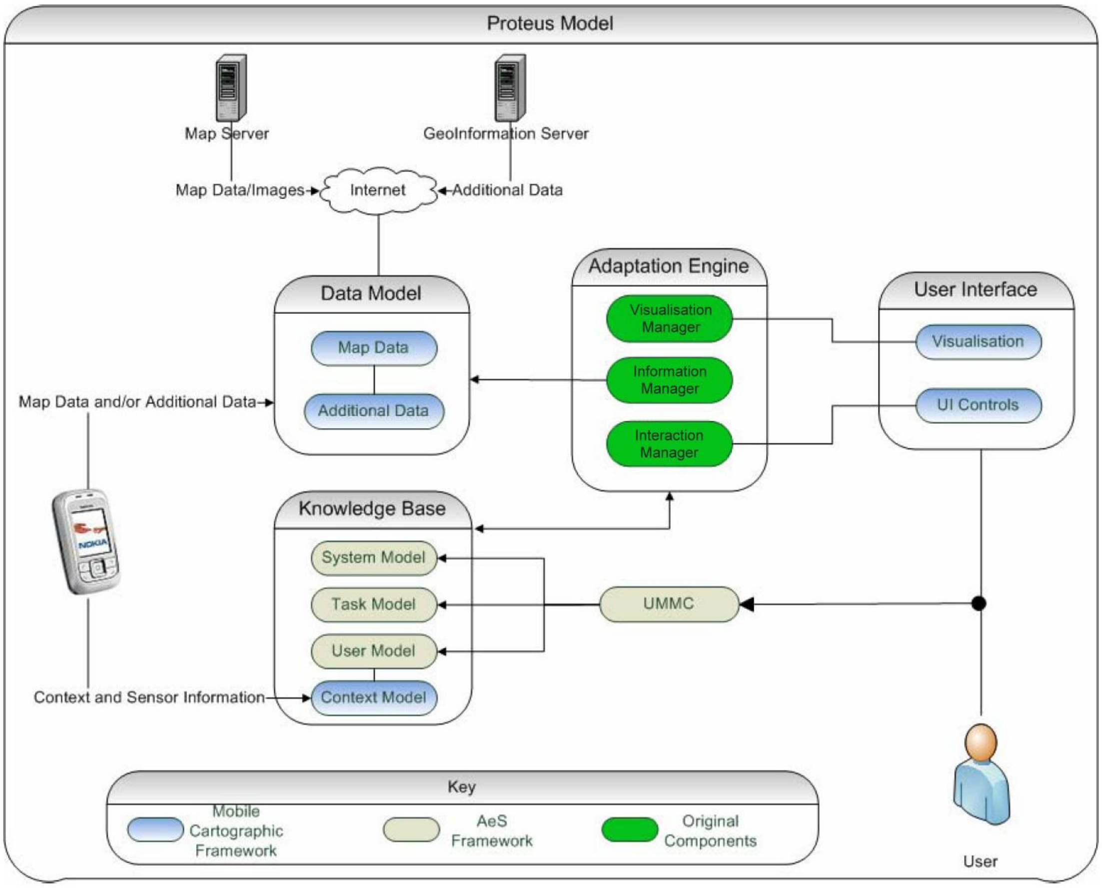


Figure 3.10: The Proteus Model for MMV systems (van Tonder and Wesson 2008)

3.4.2.2 Knowledge Base

Four sub-models are contained in the Knowledge Base that manage user knowledge relevant to the current domain, namely the System Model, Task Model, User Model and Context Model.

The S*ystem Model* acts as a central repository that holds the current state of adaptable parameters in the system that are changed by either the user or the system, in order to maintain awareness. The *Task Model* contains typical user tasks, broken down into the various subtasks required to achieve that task. It is then possible to identify a user's end goal by comparing user input actions with tasks stored in the Task Model. This allows the system to accelerate or simplify the current task the user is trying to achieve. The *User Model* stores all the knowledge acquired while a user is performing tasks, in order to build a history of user behaviour. Parameters are stored in a hierarchical list separated by task, as the user's preferences for one task may differ from his/her preferences for another. The User Model is consulted prior to performing any adaptation, in order to determine the user's preferences. This ensures that adaptations performed are aligned with the user's preference history. The *Context Model* stores context-of-use information such as the time and location. Sensor information such as GPS data is stored here as well as a unique ID to identify the current user task. This allows visualisation adaptations to occur such as adapting the zoom level according to travel speed (van Tonder and Wesson 2008).

3.4.2.3 Adaptation Engine

The Adaptation Engine contains one component for each of the three main adaptation areas (information, interaction and visualisation). Each of these components consults the Knowledge Base to ensure that actions performed match a user's behavioural preference history. The *Information Manager* filters and organises information to be displayed. The *Interaction Manager* handles changes to the user interface controls such as reordering menu items. The *Visualisation Manager* manages changes to any visual representation of information, such as the level of map detail, zoom level and map style (van Tonder and Wesson 2008).

3.4.2.4 User Monitoring and Modelling Component (UMMC)

The UMMC processes all user interaction input data and makes inferences regarding the user's preferences and behaviour. Inferred knowledge is updated in the User Model (stored in the Knowledge Base), thereby applying implicit user modelling (van Tonder and Wesson 2008).

3.4.2.5 Proteus Model Adaptation Timing

The timing of system adaptations is an important issue as it can negatively affect the usability of a mobile device if it is allowed to monopolise the limited hardware resources available. The User Model should be loaded into memory on system startup. The Adaptation Engine is invoked when the user begins a new task. For example, if a user requests to view a map, the Visualisation Manager is invoked, which consults the System and Task Models, to ensure that the map visualisation is rendered according to the user's preference history. If user interactions are logged while the system is in use, these should be written to a data file either when the system closes, or after a certain number of user actions to prevent excessive memory usage (van Tonder and Wesson 2008).

The Proteus Model successfully addresses the shortcomings of existing models by incorporating the major input variables (user, task, system and context) and supports the adaptation of objects in a MMV system for all three major adaptation types identified in Section 3.3. The Proteus Model is therefore suitable for the development of an adaptive mobile PBST, like A-POInter, that requires adaptation of the information, interaction and visualisation aspects.

3.5 Conclusion

This chapter has identified several benefits of AUIs. These include creating personalised systems by modifying functionality, task allocation or partitioning, interface transformation to reduce information overflow and providing help on using new and complex systems. It was decided that A-POInter should focus on creating a personalised system and reduce information overflow as these objectives are most relevant to the domain of mobile PBS.

Several techniques to achieve the benefits of adaptation discussed were identified. These included adaptive menus, Fisheye interfaces, Multiscale or semantic zooming and Focus+ Context techniques such as the Spiral or SnailList. Split menu techniques can be used to provide menu or list adaptation by placing the most useful options at the top of the list (based on recency or frequency). SnailLists can provide the most effective way to manipulate large lists that contain over 100 items.

Three different types of adaptation were identified (information, interaction and visualisation) and will be addressed during the development of A-POInter. The various adaptable objects commonly found in MMV systems were identified and summarised under the three main adaptation types. Examples of adaptable objects applicable to a mobile PBST and specifically to a mobile tourist guide, were identified. These adaptable objects included the encoding and amount of information displayed, level-of-detail and map features available, adaptation of functional and map interaction techniques, as well as the adaptation of the map layout, style, dimension, graphical and typographical elements to be visualised.

The development of an AUI for mobile PBS can be substantially simplified and managed by following a model-based approach. Several models were examined which support adaptation. The Proteus Model was designed to address the shortcomings of existing models for MMV systems. It supports the design of adaptive MMV systems in the three main adaptation areas and was therefore deemed most suitable for the development of A-POInter.

The next chapter examines how an adaptive interface can be developed for POInter. Several adaptation suggestions for A-POInter are proposed, based on the requirements for a mobile PBST (Chapter 2) and adaptable objects of a MMV system (Section 3.3). The results of a field study to verify the adaptive requirements for A-POInter are also discussed.

Chapter 4: Requirements Analysis

4.1 Introduction

This chapter describes a field study undertaken to verify the adaptation requirements for A-POInter[2]. The provisional adaptive requirements for A-POInter proposed in Section 4.2 were derived by combining the requirements for a mobile PBST (Chapter 2), with the adaptable objects for a MMV system (Chapter 3). Participants completed a post-test questionnaire containing the adaptation suggestions in order to verify the user requirements for A-POInter. Adaptation suggestions were divided into the three main adaptation types, namely information, interaction and visualisation. The questionnaire allowed participants to indicate how they thought A-POInter should adapt to their behaviour and preferences.

Section 4.3 describes the requirements methodology and design of the field study. Results of the questionnaire data are analysed in Section 4.4 and design decisions made. Recommendations for adaptation requirements were derived from the analysis of the results and are outlined in Section 4.5.

4.2 Provisional Adaptation Requirements

A list of adaptation suggestions for A-POInter was compiled based upon the proposed requirements for a mobile PBST (Chapter 2) and the theoretical AUI techniques and adaptable objects for MMV systems discussed in Chapter 3. These suggestions were separated into information (data) adaptation, interaction (user interface) adaptation and visualisation (presentation) adaptation and are listed in the subsections that follow.

4.2.1 Information Adaptation Suggestions

A-POInter should:

1. Adapt the starting map (geographic AOI) based on frequency, recency of use, or GPS sensor information.
2. Select the most appropriate zoom level (scale) for the map.
3. Adapt the number of POI icons (amount of search information) displayed.
4. Adjust the level-of-detail for POI search results displayed on the map.

[2] The requirements field study and analysis discussed in this chapter were published in the proceedings of SATNAC 2009 (Hill and Wesson 2009).

5. Suggest search criteria based on preference history.
6. Group POI search results according to classification or preference history.
7. Automatically filter search results.

4.2.2 Interaction Adaptation Suggestions

A-POInter should:

1. Reorder menu items and search criteria lists according to frequency and recency of use.
2. Hide menu options based on frequency and recency of use.
3. Remember preferred panning and zooming interaction techniques.
4. Implement a means to quickly zoom out in order to view the surrounding map.

4.2.3 Visualisation Adaptation Suggestions

A-POInter should:

1. Remember the preferred map style.
2. Automatically adjust the zoom level (scale) of the map according to GPS sensor information.
3. Adjust the level-of-detail and size of POI search results based on zoom level.

These adaptation suggestions were used as a basis for a questionnaire, which was used in a field study to verify the adaptation suggestions. This field study is discussed in Section 4.3.

4.3 Field Study

A field study was conducted in order to measure the extent to which participants agreed with the suggestions provided for adapting the information, interaction and visualisation aspects of POInter (Section 4.2). Analysis of these results was used to make design decisions based on the adaptation requirements for A-POInter.

4.3.1 Methodology and Design

The field study was conducted at the Nelson Mandela Bay Tourism (NMBT) information office situated at the Boardwalk (a casino, entertainment and shopping complex) in Port Elizabeth. Throughout the study, the author acted as the evaluator. Tourists visiting the office were approached and after briefly discussing the basic features of POInter verbally, were invited to participate in the study. It was explained to participants that no personal information would be collected and that they could stop the evaluation and leave at any time. Tourists who completed the study were rewarded with a R50 Boardwalk shopping voucher. Ethics approval was obtained from the NMMU REC-H Committee.

Participants were briefly demonstrated the main features of POInter by the evaluator after which they were handed the mobile device and were allowed to experiment with the system freely. Participants were then instructed to complete a set of tasks (Section 4.3.2) with POInter using a given test plan. While users were performing the tasks, they were under passive observation by the evaluator. When necessary, assistance was provided by the evaluator in order to overcome some participants' lack of familiarity with the mobile device's touch-screen interaction techniques. After completing the test plan, participants completed a post-test questionnaire. Average completion time for the entire duration of the study for each participant was approximately 15 minutes.

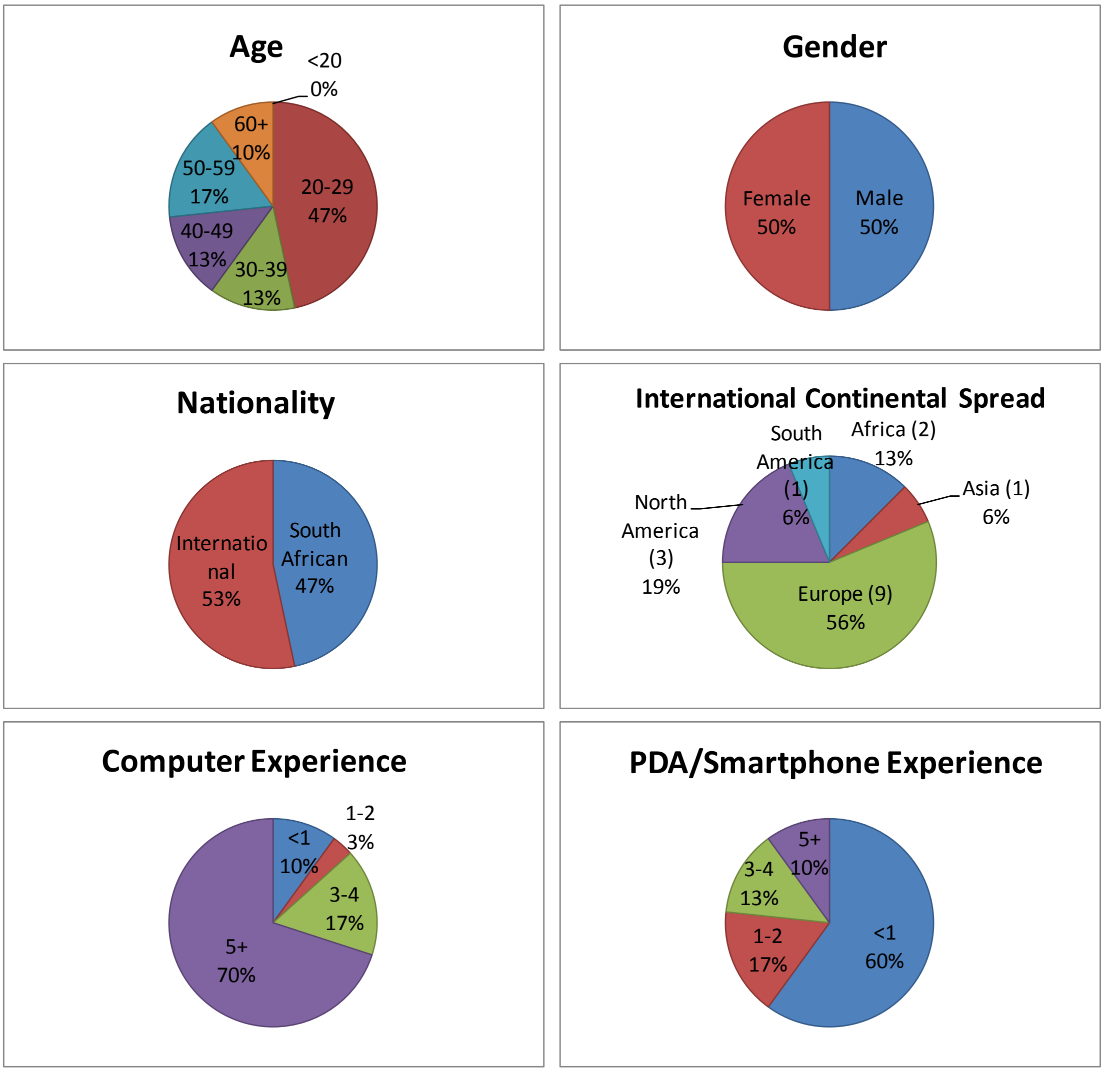


Figure 4.1: Tourist demographics (n=30)

A total of 30 tourists voluntarily participated in the study. The demographics (Figure 4.1) are summarised as follows: 14 (47%) were local tourists and 16 (53%) were international tourists spanning 5 continents; 70% were below the age of 50; there were an equal number of male and female participants; the majority of participants (70%) had over five years of general computer experience; and over half (60%) had little or no experience with smartphones and PDAs (less than one year).

4.3.2 Task Selection

The tasks in the test plan covered all the basic features of POInter. These included selecting, navigating and manipulating a map, searching for POIs (in the Accommodation category), viewing and filtering search results and interacting with specific POIs to view detailed information such as contact details. A copy of the test plan can be found in Appendix A.

4.3.3 Questionnaire Design

A post-test questionnaire containing suggestions for adapting the information, interaction and visualisation aspects of POInter (Section 4.2) was issued after each participant completed the test plan.

Participants indicated their level of agreement with the suggestions (Section 4.2) using a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). A space at the end of the questionnaire was provided to elicit other possible adaptation ideas or suggestions for improvement that were not addressed in the questionnaire. A copy of the questionnaire can be found in Appendix B.

4.4 Analysis of Results

Data capture was performed using Microsoft Excel. Sections 4.4.1-4.4.3 contain the results of the questionnaire. A measure of central tendency for each response was calculated (arithmetic mean with standard deviation, median and mode). A median score of 4 or higher was considered favourable, 2 or below as unfavourable and 3 as neutral / indecisive. For median scores of 3, the mean was examined. If the mean was less than 2.6 it was considered unfavourable, if the mean was above 3.4 it was considered favourable. These cut-points were selected as the range between 1 (Strongly Disagree) and 5 (Strongly Agree) is 4. The interpretation interval is therefore 0.8 (4 divided by 5, the number of Likert labels). Interpretation of mean values was therefore based on the following: A score of 1.0-1.8 is strongly negative, a score of 1.8-2.6 is negative, a score of 2.6-3.4 is neutral, a score of 3.4-4.2 is positive and a score of 4.2-5.0 is strongly positive. The mean was also used when the median scores between two or more suggestions were equal, in order to make a design decision.

4.4.1 Information Adaptation

Section B: Information (Data) Adaptation	Median	Mode	Mean	Std. Dev
1) POInter should remember my *most recently used* (MRU) area of interest to suggest a starting map.	4	4	4.17	0.91
2) POInter should remember my *most frequently used* (MFU) area of interest to suggest a starting map.	4	4	3.77	0.9
3) POInter should use the GPS to suggest the initial area of interest (i.e. map).	4	4	4.17	0.75
4) POInter should automatically suggest the most appropriate zoom level after searching.	4	4	3.87	0.97
5) POInter should adjust the number of POI search results shown based upon the current zoom level (e.g. show only more relevant POI results when zoomed out).	4	5	4.03	1.1
6) POInter should automatically adjust the level of detail for the map based on the current zoom level (e.g. show specific POI details such as the name at closer zoom levels).	4	4	3.93	0.87
7) When entering search criteria, POInter should suggest my MRU categories and criteria.	4	3	3.63	0.93
8) When entering search criteria, POInter should suggest my MFU categories and criteria.	4	3	3.7	0.75
9) I would like POI search results to be grouped according to certain criteria.	4	4	4.1	0.85
10) I would like POI search results to be grouped according to my preference history.	4	4	3.77	0.86
11) I would like to be able to apply a filter to view the top POIs (e.g. top 3) according to my search criteria and preference history.	5	5	4.43	0.68

Table 4.1: Information (Data) Adaptation Results (n=30)

The results of Section B of the questionnaire are given in Table 4.1. This section covered *Information Adaptation* suggestions for selecting areas of interest (maps), zoom level, specifying POI search criteria and displaying search results. These suggestions were based on the provisional requirements proposed in Section 4.2.1.

Participants preferred a *most recently used* (MRU) starting map (mean = 4.17) compared to a *most frequently used* (MFU) starting map (mean = 3.77). Most participants agreed to using the current GPS location to suggest the starting map (median = 4), however some tourists stated that when using A-POInter they would most likely be planning a trip to a location in advance and would hence would not find the current GPS location useful.

Participants agreed that A-POInter should automatically suggest the most appropriate zoom level after searching (median = 4). A-POInter should also adjust the number of

POI search results shown (median = 4) and level of detail (median = 4) based on the zoom level.

When entering search criteria, participants would prefer A-POInter to suggest MFU categories and criteria (mean = 3.7, std. dev = 0.75) as opposed to their MRU selections (mean = 3.63, std. dev = 0.93). Participants would prefer A-POInter to group search results according to certain criteria such as Accommodation (mean = 4.1) as opposed to their preference history (mean = 3.77). Lastly, participants agreed that A-POInter should automatically filter out most search results and show only the most relevant POIs according to their search criteria and preference history (median = 5).

4.4.2 Interaction Adaptation

Section C: Interaction (User Interface) Adaptation	Median	Mode	Mean	Std. Dev
1) POInter should reorder menu items based upon my MRU selections.	3	3	3.07	1.11
2) POInter should reorder menu items based upon my MFU selections.	4	3	3.6	1.13
3) POInter should hide menu options that I do not use often.	2.5	3	2.57	1.17
4) When specifying search criteria, POInter should place my MRU criteria selections at the top of the list.	4	4	4	0.79
5) When specifying search criteria, POInter should place my MFU criteria selections at the top of the list.	4	4	3.97	0.67
6) POInter should remember my preferred zooming technique (e.g. either draw a box on the map to zoom into, or use step-wise zoom in/out buttons).	4	4	4.13	0.9
7) POInter should provide a tool to temporarily zoom out of the map in order to quickly view the surrounding area.	4	4	4.27	0.64
8) POInter should remember my preferred panning technique (e.g. tap, hold and drag the map, or tap and hold directional arrows (NESW) at the edges of the map.	4	5	4.37	0.67

Table 4.2: Interaction (User Interface) Adaptation Results (n=30)

The results of Section C of the questionnaire are detailed in Table 4.2. This section covered *Interaction Adaptation* suggestions for menu adaptation, specifying search criteria and interaction techniques for zooming and panning the map. These suggestions were based on the provisional requirements proposed in Section 4.2.2.

In terms of menu item reordering, participants would prefer their MFU selections to be used (median = 4) as opposed to their MRU selections (median = 3). Participants did not want A-POInter to hide any menu options (median = 2.5). A few participants stated that the existing menu structure and number of menu items were not complex or lengthy and therefore menu adaptation was unnecessary.

When specifying search criteria, participants agreed that A-POInter should place MRU and MFU criteria selections at the top of the list (median = 4). No substantial difference could be determined between the mean scores for each and therefore a combination of both will be implemented and evaluated in A-POInter. Participants agreed that A-POInter should remember the preferred zooming technique (median = 4). Participants agreed that A-POInter should provide a means to temporarily zoom out of the map to view surrounding context (median = 4) and should remember the preferred panning technique (median = 4).

4.4.3 Visualisation Adaptation

Section D: Visualisation (Presentation) Adaptation	Median	Mode	Mean	Std. Dev
1) POInter should suggest my MRU map style (Road vs. Satellite photo vs. Hybrid).	4	4	3.7	1.18
2) POInter should suggest my MFU map style (Road vs. Satellite photo vs. Hybrid).	4	4	4.03	0.93
3) POInter should automatically adjust the zoom level according to the speed at which I am travelling.	4	4	4.33	0.71
4) When zoomed in, POInter should use a small photograph or image (e.g. depicting the actual landmark) for POI search results instead of the standard categorical icons.	4	5	4.2	0.93

Table 4.3: Visualisation (Presentation) Adaptation Results (n=30)

The results of Section D of the questionnaire are detailed in Table 4.3. This section covered *Visualisation Adaptation* suggestions for map style, zooming and graphical elements used to display search results. These suggestions were based on the provisional requirements proposed in Section 4.2.3.

Participants would prefer the MFU map style to be used (mean = 4.03, std. dev = 0.93) as opposed to the MRU map style (mean = 3.7, std. dev = 1.18). Mean values were used to make a design decision since the median scores were equal. POInter should automatically adjust the zoom level according to the speed at which the user is travelling (median = 4). Participants would like POInter to show a thumbnail image of the POI instead of the standard icon at closer zoom levels (median = 4).

4.5 Additional Qualitative Feedback

Several suggestions were received in the general section of the questionnaire, or noted during informal discussion with participants. When specifying search criteria, a few participants agreed that POInter should place MRU and MFU criteria at the top of the list, however they would not like the system to automatically select these criteria.

A few suggestions for adaptation were made by participants that were not specifically addressed by the questionnaire. These included allowing multiple personalisation profiles per user (e.g. business versus vacation) and allowing a manual override for certain system adaptations (that would otherwise be automatically adjusted according to user behaviour). For example, the ability to specify a constant zoom level when driving instead of POInter automatically adjusting the zoom level.

Three participants stated that they would like to remain in control of the system adaptation, but stated it would be annoying if POInter continually asked for confirmation to adaptation suggestions. A phased approach might therefore be applicable to system adaptation, where decision making is gradually shifted to the responsibility of the system from the initial responsibility of the user. Research has shown that system autonomy and pro-activity could lead to problems that outweigh adaptation benefits. All adaptive systems should therefore remain transparent so that the user can understand the system's inner working and predictable so that the same input always causes the same response, therefore giving the user as much 'control' over the system as possible (Pianesi, Graziola, Zancanaro and Goren-Bar 2009).

One participant suggested showing a small overview map in the corner of the map screen. This concept is related to suggestion C7 (temporarily zooming out to view the surrounding context), which participants agreed that POInter should provide (median = 4).

4.6 Adaptation Requirements

The following subsections detail design decisions, recommendations and user requirements for A-POInter based on the analysis of the results in Section 4.4.

4.6.1 Information Adaptation

Based on the results in Section 4.4.1, the following adaptation requirements were derived for A-POInter:

1. Use the MRU area of interest to suggest a starting map.
2. Automatically select the most appropriate zoom level (in order to contain all relevant POI icons onscreen) after running a search.
3. Automatically adjust the level of detail for the map based on the current zoom level (e.g. showing specific details such as the POI name at closer zoom levels).
4. Suggest MFU selections when entering search categories and criteria.

5. Automatically run a filter to show only the most relevant POI search results according to the criteria specified and their preference history (i.e. A-POInter should take preferred POIs and preferred (MFU) criteria into consideration). A-POInter should provide the ability to adjust the filter to view all search results if desired.

4.6.2 Interaction Adaptation

Based on the results in Section 4.4.2, the following adaptation requirements were derived for A-POInter:

1. Reorder menu items by placing the MFU selections at the top of the list.
2. *Do not* hide any menu items.
3. Place both MRU and MFU selections at the top of the list when specifying search criteria.
4. Remember the preferred zooming technique (based on MFU) and set it as the default (i.e. other zooming technique(s) are not available unless selected).
5. Provide a means to quickly view the surrounding map area (O*verview+Detail / Focus+Context*)
6. Remember the preferred panning technique (based on MFU) and set it as the default (i.e. other panning technique(s) are not available unless selected).

4.6.3 Visualisation Adaptation

Based on the results in Section 4.4.3, the following adaptation requirements were derived for A-POInter:

1. Always use the MFU map style.
2. Automatically adjust the zoom level according to the speed at which the user is travelling. A-POInter should zoom out when travelling faster (e.g. by car) and zoom in when travelling slower (e.g. by foot). An option to override this auto-zoom feature should be provided.
3. Provide a thumbnail image (when available) of the POI at closer zoom levels instead of showing the standard categorical icon.

4.7 Conclusions

A list of provisional adaptation requirements for A-POInter was proposed in Section 4.2. These adaptation suggestions were derived from a combination of the requirements for a mobile PBST (Chapter 2), with the adaptable objects for a MMV system (Chapter 3). The field study confirmed the need for the adaptation of POInter in all three areas of adaptation, namely, information, interaction and visualisation and verified the provisional adaptation requirements for an adaptive mobile PBST.

The results of the field study clearly showed that participants would like to use an adaptive mobile PBST. Participants generally preferred MFU selections for most adaptation parameters, including entering search criteria, filtering search results, reordering menu items, zooming and panning techniques and the selection of map style. This was to be expected since these tasks involve many repetitive user interactions with the system and should therefore increase the effectiveness of adaptations after repeated use. Recency (MRU) should be used for the selection of a starting map. This was also to be expected, as this would be useful to tourists needing to quickly access the same AOI that they used the last time. A combination of MFU and MRU items should be used for placing items at the top of the search criteria list.

Additional suggestions for adaptation (Section 4.5) included multiple user profiles as well as a need for the user to be able to maintain control over system adaptation. Manual control over certain adaptations will therefore be provided, such as the ability to override the automatic selection of a zoom level when using the GPS. It was decided not to implement multiple-user profiles in A-POInter, as the introduction of multiple user models might negatively affect ease-of-use, as well as lessen the impact and noticeability of adaptations based on frequency (MFU).

The following chapter discusses the design and development of A-POInter. The Proteus Model, introduced in Chapter 3, was used to implement adaptation requirements. Several user interface prototypes are discussed together with implications for implementation.

Chapter 5: Design and Implementation

5.1 Introduction

The design and implementation of A-POInter using a model-based design approach is discussed in this chapter. This chapter describes how the Proteus Model was used to re-engineer POInter to satisfy the adaptation requirements identified in the previous chapter. This chapter also illustrates how the architecture and individual components of the Proteus Model were used to develop A-POInter.

User interface prototypes of the different adaptation requirements identified in Chapter 4 are presented and discussed in Section 5.5, together with implications for implementation in A-POInter. The adaptation algorithms used to achieve the required adaptations are also discussed in this chapter.

5.2 Development Methodology and Implementation Tools

The development of A-POInter followed a model-based design approach. The design and implementation was based on the Proteus Model (Section 3.4.2). The Proteus Model is a model for designing an adaptive MMV system that supports a wide range of adaptations. The Proteus Model supports adaptation in three main areas, namely information, interaction and visualisation and was therefore considered as suitable for the development of A-POInter. Further discussion on the use of the Proteus Model is given in Section 5.4.

During development of A-POInter, the adaptation features were iteratively prototyped and evaluated in order to develop a usable system. Expert users reviewed the prototype and provided suggestions, where necessary, to improve the usability of the prototype designs.

POInter (Hill 2007) was designed and developed using Microsoft Visual Studio 2005, with the Windows Mobile 5.0 SDK. It targeted Windows Mobile 5.0 Pocket PC touchscreen devices, with a screen resolution of 240x240 using the .NET Compact Framework v2.0. A-POInter was developed using Microsoft Visual Studio 2008, with the Windows Mobile 6 Professional SDK. It targets Windows Mobile 5.0 PocketPC, Windows Mobile 6.0 Professional (or later) touchscreen devices, with a 240x320 (QVGA) screen resolution. It requires the .NET Compact Framework v3.5.

During initial development, the emulator provided with the SDK was utilised to develop and test prototypes for usability and functionality. It was, however, necessary to test the system using an actual device that supported GPS functionality and the ability to effectively evaluate user interaction and satisfaction using a touchscreen. For this reason, a HTC TyTN II Windows Mobile 6.0 Professional touchscreen device was used to further test, implement and evaluate prototypes. The device was upgraded to Windows Mobile 6.1 Professional via an official firmware upgrade. A 2GB storage card was used in the device to facilitate the storage of cached map images that are downloaded when using A-POInter.

5.3 Adaptation Algorithms

Several adaptation algorithms exist for managing the generation of stored preferences and controlling the information, adapting the user-interface and managing changes to the presentation or visualisation of information. For information adaptations that require the use of a preferred item, the most frequently used (MFU) or most recently used (MRU) items are commonly used (Reichenbacher 2004; Gajos *et al.* 2006; McGrenere *et al.* 2007; Findlater, McGrenere and Modjeska 2008). Frequency can be determined by counting how many times each item is selected and storing this information in a user model so that the MFU items can be retrieved.

5.3.1 Base Adaptive Partitioning Algorithm

Findlater *et al.* (2004) developed an adaptive partitioning interaction adaptation algorithm to facilitate an adaptive split-menu (Figure 5.1). Of the four items reserved in the top partition, the algorithm places two items based on frequency (MFU items) and two items based on recency (MRU items). Frequency is captured simply by logging a count of how many times each menu item was selected. Other explicitly specified options and selections made in a system can be logged and preferences inferred implicitly. The system can then interpret and use these data logs and patterns to adapt in various ways.

```
(initially: item.frequency = item.recency = 0)

selectedItem.frequency++
selectedItem.recency = 0
for each remaining item[i] in the menu
    item[i].recency++
    if the selected item is in the top partition already
        do nothing
   else
        leastRecItem =  least recently used of the recency items
        leastFreqItem = least frequent of the frequency items
        if leastRecItem.frequency < leastFreqItem.frequency
             move leastRecItem to bottom partition
        else
             move leastFreqItem to bottom partition
             set leastRecItem.type = frequency item
        move selectedItem to top partition of menu
        set selectedItem.type = recency item
```

Figure 5.1: Adaptive algorithm for a split-menu (Findlater and McGrenere 2004)

5.3.2 Threshold Value-Based Adaptations

Several algorithms have been devised to provide adaptive functionality based upon threshold values. For example, to control a visualisation adaptation (Section 3.3.3), a simple IF-Statement algorithm as shown in Figure 5.2 could be used (Reichenbacher 2003). A filter method is applied to reduce the amount of objects displayed on a map depending on the speed at which the user is travelling.

```
IF (speed < 10 km/h)
        THEN (filter, 90%);
ELSE IF (speed >= 10km/h AND speed < 50 km/h)
        THEN (filter, 60%);
```

Figure 5.2: Level-of-Detail visualisation filtering algorithm (Reichenbacher 2003)

There are many instances where a threshold value could be used in a mobile PBST, such as selecting the most appropriate zoom level after executing a search and controlling the level of POI detail according to the current map scale (zoom level) (Reichenbacher 2003).

5.4 The Proteus Model

The following subsections describe which aspects of the Proteus Model were used to develop A-POInter, as well as any changes necessary and difficulties encountered.

The Proteus Model architecture as implemented in A-POInter is shown in Figure 5.3. The design of A-POInter in terms of the components of the Proteus Model (Section 3.4.2) is described in the subsections that follow.

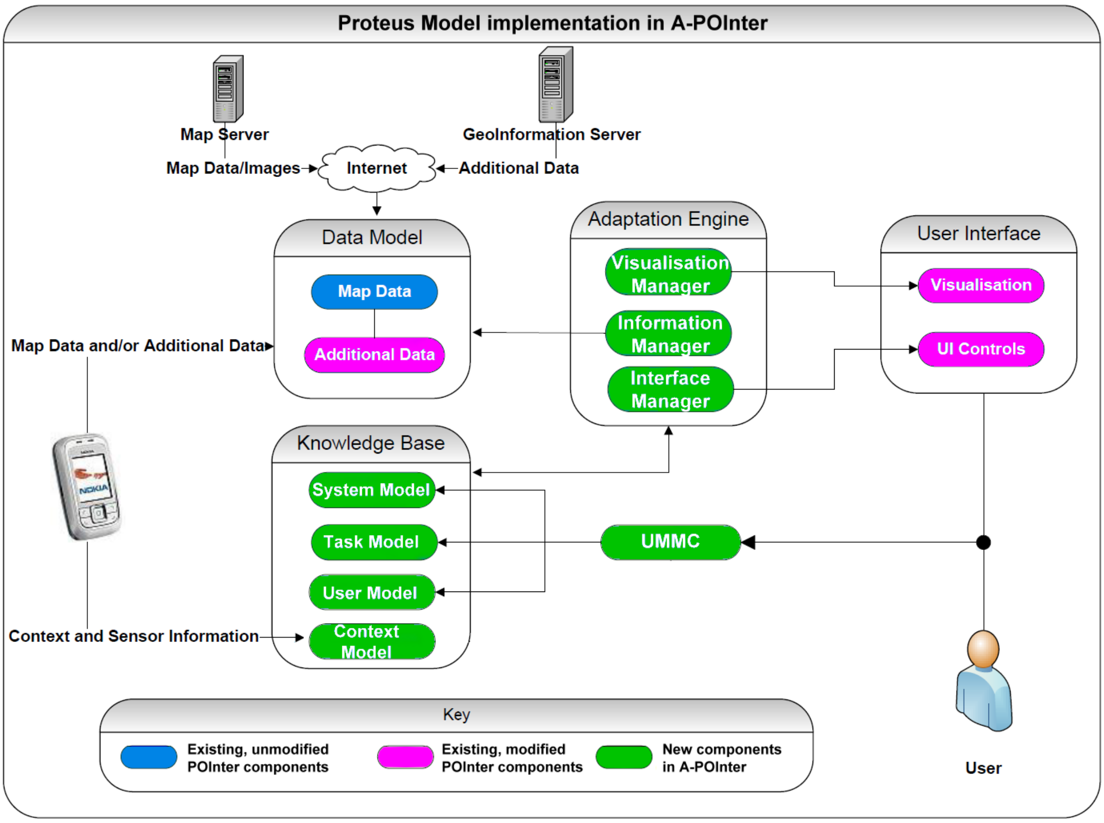


Figure 5.3: The Proteus Model for MMV systems as implemented in A-POInter

5.4.1 Data Model

In POInter, the map data is streamed as image tiles from Microsoft Bing Maps (formerly known as Microsoft Live Maps) (Microsoft 2009) and cached on the device. Additional data includes the offline XML storage of POI data such as Accommodation details (Hill and Wesson 2008). As the Proteus Model contains a clear separation of adaptation functionality from system functionality, few alterations to the Data Model in A-POInter were necessary. The only modification required was an additional field (an XML element) to store the file path of a thumbnail image of the POI.

5.4.2 Knowledge Base

5.4.2.1 System Model

The System Model holds the current status of adaptable parameters in A-POInter, which are changed during runtime either automatically by the system or manually by the user. An extract of the structure of the class is shown in Figure 5.4.

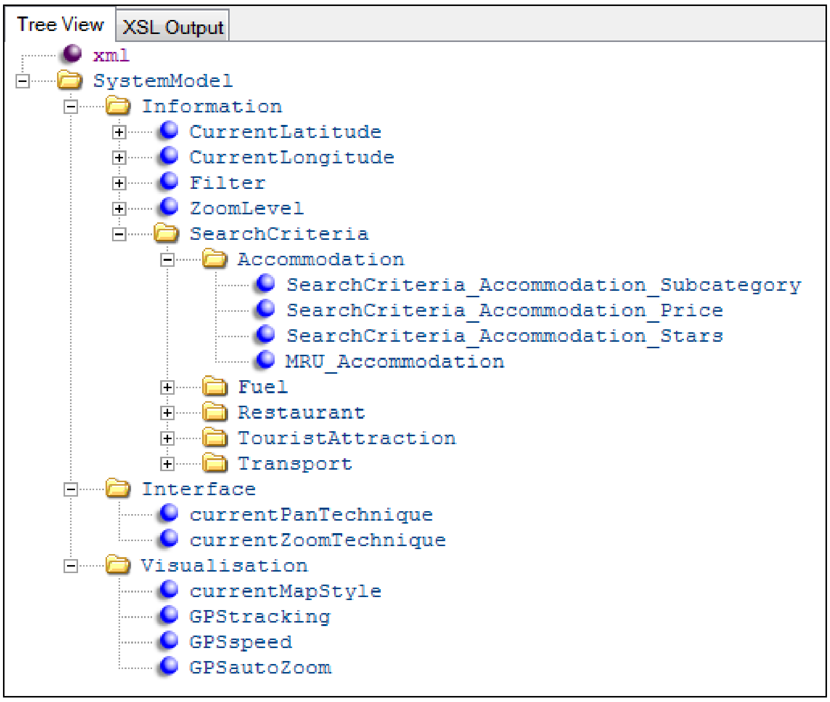


Figure 5.4: System Model structure (extract) in A-POInter

The adaptable attributes (shown as blue XML elements in Figure 5.4) were implemented as variables in a single class in A-POInter. A brief explanation of the values stored in each attribute is given in Table 5.1.

5.4.2.2 Task Model

The Task Model contains a hierarchical list of typical user tasks in A-POInter. Monitoring user actions allows A-POInter to identify the current task by comparing these actions to the subtasks listed in the Task Model. Once the current task has been identified, the Context Model is updated to store the current task ID. Key user tasks in A-POInter include selecting an AOI (starting map), searching for POIs, manipulating the map by zooming, panning or changing the map style and using the GPS sensor to track the current user location.

The Task Model was hard-coded into A-POInter for maximum efficiency. The structure of the Task Model is shown in Figure 5.5. Tasks are decomposed into the different subtasks that each task contains.

Each time a user invokes a subtask, A-POInter stores the associated parent task's identifier in the Context Model (see Section 5.4.2.4). This value is used to identify which set of parameters should be modified in the System Model and the User Model.

System Model Attribute	Value
Information	
CurrentLatitude	Stores the current map viewport latitude. Modified when panning, viewing a new map AOI, etc.
CurrentLongitude	Stores the current map viewport longitude.
Filter	Stores the preferred number of POI results a user would like visualised after a search.
ZoomLevel	Stores the current zoom level of the map.
SearchCriteria_Accommodation_Subcategory	Stores a count of how many times each subcategory (e.g. Hotel) was used, in a Dictionary (key/pair) collection.
SearchCriteria_Accommodation_Price	Stores a count of how many times each price range (e.g. RRR - Moderate) was used, in a Dictionary (key/pair) collection.
SearchCriteria_Accommodation_Stars	Stores a count of how many times each stars range (e.g. 3 Stars) was used, in a Dictionary (key/pair) collection.
MRU_Accommodation	Stores the MRU accommodation subcategory.
Interface	
currentPanTechnique	Stores the current pan technique.
currentZoomTechnique	Stores the current zoom technique.
Visualisation	
currentMapStyle	Stores the current map-style.
GPStracking	Stores whether GPS tracking is currently ON or OFF.
GPSspeed	Stores the current speed at which the user is travelling.
GPSautoZoom	Stores whether the manual override for GPS automatic zooming is ON or OFF.

Table 5.1: System Model attributes and values (extract)

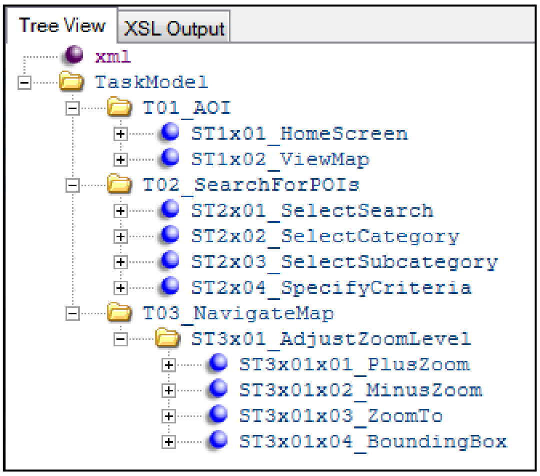


Figure 5.5: Task Model class representation in A-POInter

5.4.2.3 User Model

The User Model in A-POInter stores a history of user preferences separated by task and type of adaptation. The User Model is stored in XML format when A-POInter is not running (Figure 5.6). The User Model class contains methods to save and load the model structure and contents to XML. A brief explanation of the values stored in each attribute is given in Table 5.2.

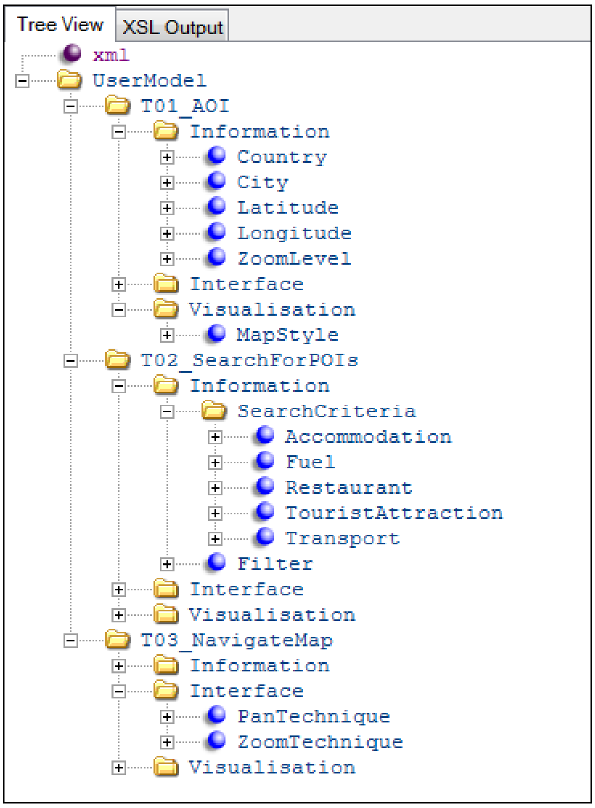


Figure 5.6: Extract of the User Model used in A-POInter

User Model Attribute	Value
T01_AOI	
Information	
Country	Stores the MRU Country (from the map selection screen).
City	Stores the MRU City (from the map selection screen).
Latitude	Stores the MRU map's latitude.
Longitude	Stores the MRU map's longitude.
Zoom Level	Stores the MRU map's zoom level.
Visualisation	
Map Style	Comma delimited string containing a count of the number of times the map style Road, or Hybrid has been used
T02_SearchForPOIs	
Information	
Search Criteria	
Accommodation	Comma delimited string containing a count of the number of times each accommodation criteria type was used.
Filter	Stores the preferred number of POI results a user would like visualised after a search.
T03_NavigateMap	
Interface	
PanTechnique	Comma delimited string containing a count of the number of times each panning technique was used.
ZoomTechnique	Comma delimited string containing a count of the number of times each zooming technique was used.

Table 5.2: User Model attributes and values (extract)

5.4.2.4 Context Model

The Context Model in A-POInter stores GPS sensor-based information in order to track user location (pan the map) and automatically adjust the zoom level and level-of-detail according to the user's current travelling speed. The current task ID is stored, so that when a new user task is detected (and the Context Model updated), the relevant manager components in the Adaptation Engine can be invoked. The structure of the Context Model is shown in Figure 5.7.

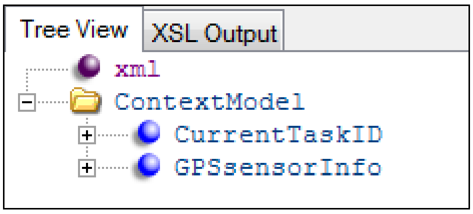


Figure 5.7: Context Model class representation in A-POInter

The Context Model was implemented as a single class in A-POInter. String values for the current task identifier, as well as raw GPS string data (containing latitude, longitude, speed, etc) that is obtained continuously when the GPS sensor is on, are updated and stored in this class for retrieval when required.

5.4.3 Adaptation Engine

The Adaptation Engine contains one sub-component for each of the three main adaptation types (information, interaction and visualisation). Each of these sub-components was implemented as a separate class in A-POInter. These subcomponents consult the Knowledge Base to ensure that the actions performed match the user's preference history. Some methods from existing classes in POInter were extracted and placed in the newly created classes to conform to the Proteus Model as well as improve reuse of code.

The *Information Manager* filters and organises information to be displayed. The Information Manager is responsible for adapting the starting map, zoom level, level-of-detail, search criteria selected and filtering search results. The *Interaction Manager* handles changes to the user-interface controls, such as reordering criteria lists in A-POInter. The *Visualisation Manager* manages changes to any visual representation of information, such as the zoom level and map style in A-POInter. Specific adaptations handled by the Adaptation Engine are discussed in Section 5.5.

5.4.4 User Monitoring and Modelling Component (UMMC)

The UMMC in A-POInter accepts user interaction input data (such as the selection of search criteria) and converts this into knowledge by making inferences regarding the user's preferences and behaviour. Implicit user modelling is used in A-POInter to derive the User Model. As a user continues to use the system, his/her preferences are processed by the UMMC and stored in the System and User Model (Section 5.4.2).

Interaction data is used to identify the user's current task from the Task Model, after which the Context Model is updated to store the current task ID. This allows the necessary manager components to be invoked in order to perform the required adaptations (Section 5.4.3).

The UMMC controls reordering of criteria lists, tracking of MRU and MFU selections and items and various other threshold value based adaptations such as adjusting the zoom level according to the current speed.

5.4.5 Adaptation Timing

It was important that the timing of system adaptations was managed correctly so that they did not monopolise the limited mobile device hardware capabilities. Reading and writing to and from XML files can take a long time. To prevent excessive memory usage and long periods of delay, the User Model is read from XML at system startup and saved on program exit. On program startup, attribute values are read in from the User Model XML file and used to initialise attribute values in the System Model class. Comma-delimited strings from User Model attributes (Table 5.2) are parsed into individual attributes into the System Model (Table 5.1). On program exit, attribute values from the System Model are used to update the User Model and write the User Model back to XML.

The Adaptation Engine is invoked when the user begins a new task. For example, if a user requests to view a map, the Visualisation Manager is invoked, which consults the System and Task Models, to ensure that map visualisation is rendered according to the user's preference history.

The System Model is updated as often as required during normal program operation. For example, whenever a new map visualisation is processed (e.g. after specifying criteria to search for POIs), the current latitude, longitude, number of POI results on screen (filter), zoom level and preferred search criteria are updated in the System Model.

5.5 Adaptation

The design and implementation of the adaptive features in A-POInter is discussed in the following sections, separated into the three adaptation types, namely Information, Interaction and Visualisation adaptation.

5.5.1 Information Adaptation

5.5.1.1 Automatic Selection of a Starting Map

On program startup in POInter, a user is taken directly to the "Home Screen" (Figure 5.8). In A-POInter a user is presented with a splash screen (Figure 5.9) on startup while the system is loading XML resources into memory.

Figure 5.8: Home Screen (240x240 pixels) in POInter (left) and the Home Screen (240x320 pixels) in A-POInter (right)

Upon load completion, a user is taken directly to his/her MRU map (Figure 5.9). If no MRU map exists (e.g. the system is being used for the first time, then the Home Screen will be displayed instead.

In POInter tapping "View Map" from the Home Screen would take the user to a screen for selecting a map, or to detect the starting map using GPS. In A-POInter, tapping "View Map" from the Home Screen (Figure 5.8) will immediately visualise the MRU map. To change the map, a user must use the Main Menu (Figure 5.9) and select "Menu > Change Map", after which he/she will be presented with the controls to select a desired starting map (Figure 5.10). If no MRU map exists, tapping "View Map" will take the user to the map selection screen (Figure 5.10).

Figure 5.9: A-POInter Splash Screen (left) and MRU map with main menu (right)

Figure 5.10: Changing the AOI (map) in A-POInter

In order to visualise the MRU map, four values are retrieved from the System Model: *CurrentLatitude*, *CurrentLongitude*, *ZoomLevel* and *currentMapStyle*. On program load, these values are copied from the User Model into the System Model. Once the map has loaded, further panning of the map will update the MRU latitude, longitude and zoom level attributes.

5.5.1.2 Automatic Filtering of POI Search Results

POInter used a manual filter slider to control the number of search results visualised. The percentage based slider indicated the percentage ranking of the search results. POIs were ranked according to the extent to which they matched the user's specified criteria (Figure 2.7).

After calculating a set of search results, A-POInter will automatically set the filter percentage to a value that will visualise as many results as the user's desired number of search results. After calculating and setting this optimal filter level, the optimal zoom level is calculated (as discussed in Section 5.5.1.5) and the map is zoomed in or out accordingly.

The algorithm used to adjust the filter percentage is 'adaptive' in its nature. Upon using A-POInter for the first time and running a search, the adaptive filter is set to produce as close as possible to 5 results in the candidate set of results. If after visualisation, the user manually changes the filter slider, the 'preferred' filter level is learned. For example, if the old filter was set to produce 5 results and the new filter level produces 15 results, the new "*Filter*" attribute is calculated as an average of the old and new values (e.g. [5+15] / 2 = 10). An average was used in order to maintain a smooth adjustment of the filter over time.

5.5.1.3 Automatic Adjustment of the Level of Detail

In POInter, only the categorical icon for POI search results was shown at all zoom levels (Section 2.4). In A-POInter, the level of detail for POI data changes depending on the current zoom level. At higher (closer) zoom levels extra details such as the POI name are shown next to the POI icon. At zoom level 8/10, the POI name (limited to 8 chars) is shown underneath the POI icon (Figure 5.11), in order to minimise occlusion. At zoom level 9/10 and 10/10, the full POI name (unrestricted length) is shown underneath the POI icon (Figure 5.11).

Figure 5.11: Zoom level 8 (left) and zoom level 9 (right) in A-POInter showing level-of-detail adjustments

Early prototypes of A-POInter displayed a thumbnail image of the POI at the maximum zoom level (i.e. zoom level 10/10). User evaluation feedback, however, revealed that displaying the thumbnail images produced a high level of occlusion when multiple POIs were visualised (Figure 5.12). The thumbnail images were therefore removed from the map view display and included in the "View POI Details" screen (Figure 5.13).

Figure 5.12: High levels of occlusion (left) when thumbnail images are shown in an early prototype, vs. acceptable levels of occlusion (right) in A-POInter at the maximum zoom level

Figure 5.13: The thumbnail image is available on the "View POI Details" screen in A-POInter

5.5.1.4 Suggesting MFU Search Criteria

After a user initiates a search for a category of POIs in A-POInter, the system automatically selects the MFU criteria (Figure 5.14). Figure 5.14 also shows the adaptation of list reordering, using the Base adaptive partitioning algorithm (Section 5.5.2.2). The two MFU criteria are selected, but the MRU (3rd item) is not automatically selected. In this case, the MFU classification (subcategory), Price ranges (Accommodation and Restaurant only) and Stars (Accommodation only) will be automatically selected when specifying a new search, based on the user's preference history.

Figure 5.14: A-POInter automatically selects MFU criteria and places the 2 MFU items and 1 MRU item at the top of the list

On program load, the relevant comma-delimited string is parsed from XML into the User Model (Table 5.2). The comma-delimited string contains counts for the number of times a specific criteria field has been used as well as a unique ID for the MRU subcategory. The string is parsed and split into individual values stored in the System Model using a standard dictionary collection. Unique key/value pairs (string/integer) are used to store MFU/MRU values.

Each time the user runs a search, the MFU count for the relevant criteria are incremented and the MRU subcategory is stored in the dictionary. On program exit, the dictionary values are written back to a comma-delimited string and written to XML.

5.5.1.5 Automatic Selection of an Appropriate Zoom Level

After specifying search criteria in A-POInter, a user can select the "View Results" icon to view search results on the map. Previously in POInter, this would visualise search results at the same zoom level that was used prior to the search criteria being specified (Figure 2.7). This could result in too many or too few search results being displayed on the screen at the current zoom level. In A-POInter, after a user selects "View Results", the zoom level is adjusted in order to ensure that a useful number of POI search results are visualised in the current map (Figure 5.15).

After running a search, A-POInter automatically pans the map to the nearest POI search result and zooms to the maximum zoom level (level 10). A-POInter then zooms out step-by-step (with an approximately 1 second pause in-between each zoom level / map refresh) until at least five POIs are shown onscreen (this count includes nearby POIs visualised as Halos). Informal user testing revealed that five POIs were the most useful number of results to be initially shown after running a search. A-POInter will not zoom out beyond zoom level 6 (city-level), in the case where too few POI results are produced by the search.

Figure 5.15: A zoom level too close (high) visualises too few search results in POInter (left). In A-POInter, the zoom level is automatically adjusted (right) to reveal a more useful number of POI results (min = 5)

5.5.2 Interaction Adaptation

5.5.2.1 Reordering and Hiding Menu Items

Hiding main menu items in A-POInter was not implemented in accordance with the results of the requirements analysis. Additionally, it was decided not to reorder menu items in A-POInter, as early prototype evaluation revealed that participants found it confusing and frustrating when menu items were moved. Prototype evaluation revealed that the menu items in A-POInter were not complex or long enough to warrant adaptation.

5.5.2.2 Reordering Criteria List Items

A modified version of the "base adaptive" partitioning algorithm (Findlater and McGrenere 2004) that combines both MRU and MFU selections was used (Section 5.3.1). The first few positions in a menu or list are reserved for the user's most likely (preferred) selections (Figure 5.14). As the criteria lists in A-POInter are not lengthy, the MRU and MFU items were moved and not copied (duplicated) as the original algorithm specified.

The algorithm used in A-POInter uses the following rules to govern adaptation:

1. The top section ("Preferred") contains two MFU selections and the MRU selected item.
2. If duplication occurs between MRU and MFU items, the third MFU item is included in order to always retain three unique items.
3. Items appearing in the top section will be ordered as they would appear in the bottom section (i.e. alphabetically).

To reorder the criteria lists, the MFU and MRU values are retrieved from the dictionary stored in the System Model. The two MFU items are placed in the top section (under the heading "Preferred") followed by the MRU item (or 3rd MFU). Each time an item is placed, it is marked in memory as being 'used'. Unplaced (unused) items are then simply placed one beneath the other in the lower section of the form, under the heading "Other", in alphabetical order (Figure 5.14).

5.5.2.3 Methods to Quickly View Surrounding Map Area

A-POInter utilises raster map images and discrete (step-wise) zoom levels. It was therefore decided not to implement this feature, as it was not possible to transition between two different raster map images (due to the discontinuation of zoom levels), without slowing the application down to unacceptable levels.

5.5.2.4 Automatic Activation of Preferred Panning Technique

In order to provide adaptation involving a preferred panning technique, at least two panning techniques were required. The pen drag panning technique was used in POInter (Section 2.3). A new panning technique was added in A-POInter, using North-East-South-West (NESW) arrows (Figure 5.16). Tapping or holding a directional arrow at the edge of the map screen pans the map in that direction (until the stylus is lifted). While the NESW pan technique is active, a user is unable to pan the map using the alternative pen drag panning technique.

Figure 5.16: The NESW arrows panning technique (left) vs. Pen Drag panning technique (right)

A user can toggle the current panning technique by either selecting the desired technique from the menu, or by toggling the active technique by tapping the shortcut that appears on the toolbar. The NESW icon  is displayed when the NESW technique is active and the pen drag icon  is displayed when the Pen Drag technique is active.

Whenever the panning technique is changed, the technique currently activated is saved to the System Model. On program exit, the MFU count in the User Model is incremented and written to XML. On program startup, the MFU technique is activated and the screen controls updated accordingly.

5.5.2.5 Automatic Activation of Preferred Zooming Technique

In order to provide adaptation involving a preferred zooming technique, at least two zooming techniques were required. Zooming in POInter was achieved by tapping a plus or minus icon (Section 2.3). A new zooming technique was added in A-POInter, which allows the user to 'draw' a bounding box on the map in order to zoom and pan into that area (Figure 5.17).

Figure 5.17: Bounding box technique active (left) and drawing a box using the stylus (right)

A user can toggle the current zoom technique by either selecting the desired technique from the menu, or by toggling the active technique by tapping the shortcut that appears on the toolbar. The bounding box icon  is displayed when the bounding box technique is active and the step wise icon  is displayed when step wise zooming technique is active. In addition, the zoom-in icon is adjusted to show either the draw bounding-box icon  or step-wise plus icon .

To zoom in with the bounding-box technique, a user taps the draw bounding-box icon, which turns green to indicate that a box can now be drawn on the screen (Figure 5.17). As the user drags the pen across the screen, a box is displayed in orange. Upon releasing the stylus, A-POInter pans to the centre of the box that was drawn and zooms in a level.

Whenever the zooming technique is switched, the technique currently active is saved to the System Model. On program exit, the MFU count in the User Model is incremented and written to XML. On program startup, the MFU technique is activated and the screen controls updated accordingly.

5.5.3 Visualisation Adaptation

5.5.3.1 Automatic Selection of Preferred Map Style

It is important for a mobile PBST to offer both road and satellite map imagery (Section 2.4). Upon program startup and loading the MRU map, the MFU map style (road map or hybrid) is selected. The MFU counts obtained from the User Model are copied into the System Model and the larger MFU count selected.

The MFU map count is only incremented on program exit. In other words, the map style that was last used has its counter incremented by one and this value is saved back into the User Model.

5.5.3.2 Automatic Tracking and Zooming According to GPS Sensor Information

A user can toggle GPS tracking ON or OFF in A-POInter by selecting the main menu option, or by toggling the icon found on the toolbar. When the GPS tracking is off (Figure 5.18), the GPS off icon is shown on the toolbar . Tapping this icon will turn the GPS on and change the icon to show that it is on . Once the GPS position has been triangulated, an icon of a car is placed on the map indicating the user's exact position (Figure 5.18). The map is automatically centred on the user's position and the zoom level automatically adjusted according to the speed at which the user is travelling. While the GPS is activated, raw sensor data (current latitude, longitude and speed information) is passed to and updated in the Context Model. Every 3 seconds, GPS speed data is sampled, converted from knots into kilometres per hour and stored in the System Model so that it can be used for automatic zooming.

Figure 5.18: GPS tracking OFF (left) and GPS tracking ON (right)

Threshold value-based adaptation (Section 5.3.2) was used to determine the zoom level to be selected. For example, if the current speed is between 0 km/h and 10 km/h (a walking/running pace), the zoom level is set to 10 (maximum). If the current speed is between 10 km/h and 60 km/h (normal city driving), the zoom level is set to level 9 and so on. The ranges selected correspond with the automatic zooming ranges used in existing mobile navigation systems and were compared to the Garmin Mobile XT for Windows Mobile (Garmin 2009) and the Mio C320 GPS car navigation system (Mio 2009). A manual override for zooming is available while the GPS is on. To toggle automatic zooming, a user can select the option from the main menu.

5.6 Discussion

The implementation of A-POInter discussed in this chapter using the Proteus Model, provides evidence to support that there are several benefits to be obtained from using a model-based design to design an adaptive MMV system. These benefits include the specification of the system architecture and key components (Section 5.4.1). Referral to the original model throughout development reduced the cognitive load associated with maintaining a mental model of how the system fitted together and therefore made it easier to develop the system.

The development of A-POInter also demonstrated that the Proteus Model can be used to successfully implement the adaptation requirements for an adaptive MMV system. The successful implementation of A-POInter provides evidence to support that the Proteus Model can be used to re-engineer an existing MMV system and not only to develop new adaptive MMV systems. The clear separation of AUI functionality from system functionality in the Proteus Model allowed the implementation of A-POInter to proceed with minimal alterations to the source code of the existing POInter system.

The Proteus Model as implemented in A-POInter was shown in Figure 5.3. The Data Model and User Interface (visualisation and controls) already existed in POInter. An additional field was included in the Data Model to enable thumbnail images to be stored (Section 5.4.1). Additional changes to the UI included enhancements to the visualisation (e.g. level-of-detail for POIs) as well as the inclusion of new UI Controls (e.g. new panning and zooming techniques). New components that were incorporated into POInter included the Knowledge Base (Section 5.4.2), the Adaptation Engine (Section 5.4.3) and the UMMC (Section 5.4.4). The implementation of these new components in A-POInter was straightforward. Each of these components and their sub-components was implemented as a separate class in A-POInter. The Proteus Model showed how these components and classes should interlink, which facilitated the bridge between design and implementation of A-POInter.

5.7 Conclusions

This chapter discussed the design and implementation of A-POInter. Several algorithms and methods to enable system adaptation were identified in Section 5.3. These included a partitioning algorithm to enable the dynamic menus required for A-POInter interaction adaptation and threshold value-based algorithms for adapting visualisation adaptations in A-POInter.

Information adaptations successfully implemented in A-POInter include the automatic selection of a starting map, automatic filtering of POI search results, automatic adjustment of the level of map detail, automatic selection of preferred search criteria and the automatic selection of an appropriate zoom level (Section 5.5.1). Interaction adaptations successfully implemented in A-POInter included the reordering of criteria lists based on frequency and recency and automatically activating preferred panning and zooming techniques (Section 5.5.2). Visualisation adaptations successfully implemented in A-POInter included the automatic selection of a preferred map style and the automatic tracking of position and map zooming according to GPS sensor information (Section 5.5.3).

A few adaptations listed in the adaptation requirements (Section 4.5) were not implemented in the final version of A-POInter, namely reordering of main menu items, providing a thumbnail image for a POI instead of the categorical icon on the map and providing a means to quickly view the surrounding map area. These adaptation features were removed after user testing revealed that reordering main menu items was confusing and visualising a POI thumbnail on the map introduced undesirable levels of occlusion. Providing a means to quickly view the surrounding map area was not implemented due to the inadequate hardware capabilities of the mobile device. Smooth raster image rescaling proved to be unattainable with the limited CPU power and memory resources available. All other adaptations were successfully implemented in A-POInter.

The Proteus Model was successfully utilised to enable the implementation of A-POInter. The existing Data Model and User Interface in POInter were modified and a Knowledge Base, Adaptation Engine and UMMC successfully added during the implementation of A-POInter. The successful implementation of A-POInter provides substantial evidence that the Proteus Model is flexible enough to support the design of adaptive MMV systems in the future. This implementation also served to identify several benefits of using a model-based design approach, including simplicity and efficiency.

The following chapter discusses the evaluation of A-POInter which involved an international field study to determine the usability benefits of incorporating an AUI into a mobile PBST.

Chapter 6: Evaluation

6.1 Introduction

The previous chapter outlined the design and implementation of A-POInter. This chapter describes techniques for evaluating AUIs and mobile usability, the challenges associated with evaluating AUIs and mobile applications, various evaluation instruments available and metrics applicable to the evaluation of adaptive MMV systems such as A-POInter.

An international field study was conducted to determine the usability benefits of A-POInter. The results of this field study are discussed and analysed. The chapter concludes with a discussion of the results and includes a comparison to the initial adaptation requirements identified in Chapter 4.

6.2 Evaluation Techniques

6.2.1 Evaluating AUIs

Evaluation is a fundamental aspect during the development of an AUI (Gena 2005). According to Alvarez-Cortez *et al.* (2007), a limited number of empirical evaluations of adaptive systems has been performed and more research and evaluation is needed to determine if an adaptive interface is measurably superior to a static interface. Several problems exist with the evaluation of AUIs which contribute to their apparent lack of evaluation during development. These problems include difficulties in attributing cause, large variances in user characteristics or behaviour affecting statistical significance, difficulty in defining the effectiveness of adaptations due to adaptation timing and difficulty in recruiting large numbers of participants (Masthoff 2003).

It is important to evaluate the effectiveness of the adaptation. Most evaluations of adaptive systems compare the adaptive system with a current non-adaptive system. The AUIs researched by Gajos, Wobbrock and Weld (2008) were evaluated using two methods, namely active elicitation and example critiquing. With active elicitation, participants were presented with queries showing pairs of user interface fragments and were asked which (if either) they preferred. Participants had the opportunity during example critiquing to provide qualitative feedback and suggest improvements to the interfaces.

An adaptive system has the potential to suffer from usability drawbacks such as a perceived loss of user control due to unexpected adaptation (Alvarez-Cortes *et al.* 2007). The benefits of an adaptive system should substantially outweigh any potential usability problems (Pianesi *et al.* 2009). An overview of usability challenges for user-adaptive systems is shown in Figure 6.1. In Figure 6.1, solid arrows depict positive influences, while dashed arrows represent negative influences. Usability goals corresponding to generally desirable properties of interactive systems include: predictability and transparency, controllability, unobtrusiveness, privacy and breadth of experience. Additionally, it is important not to restrict empirical studies to modelling accuracy (Jameson 2002).

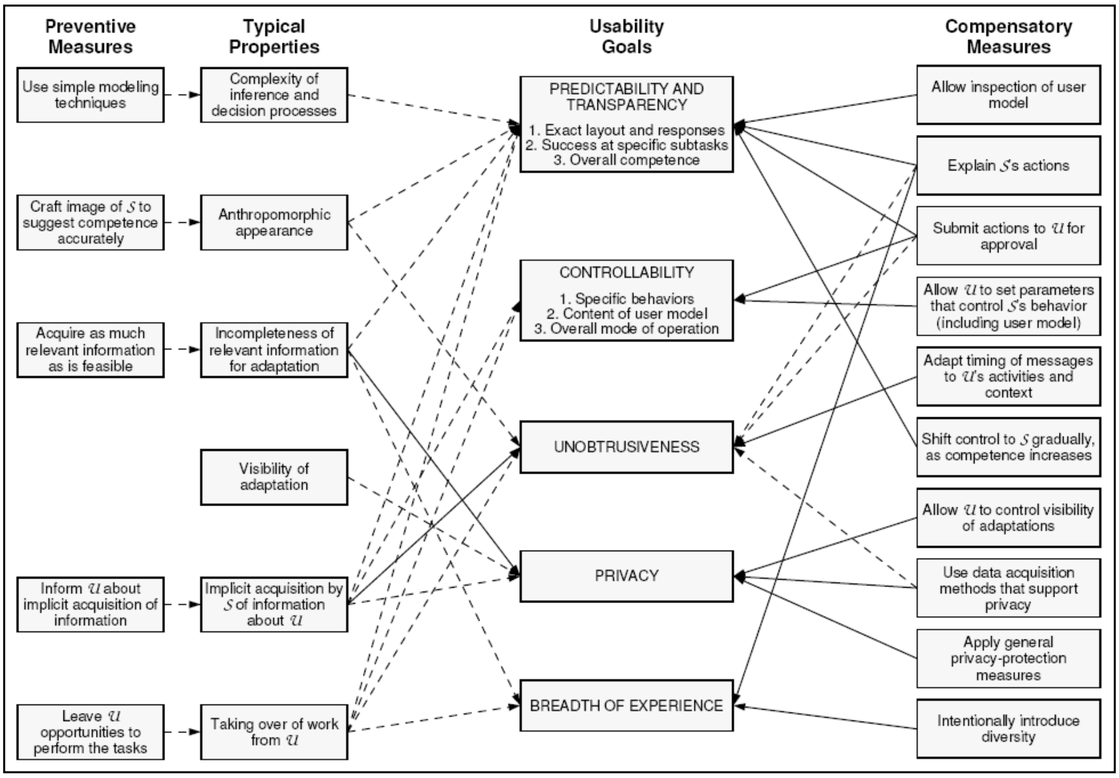


Figure 6.1: Usability challenges for adaptive systems. U=user; S=System (Jameson 2002)

A-POInter achieves the usability goal of unobtrusiveness as it does not explain the system's actions or submit actions to the user for approval. This could, however, negatively impact transparency as the user might not always understand the system actions taken and additionally removes some perceived user control over the system. A-POInter should, however, be predictable as relatively simple modelling techniques and algorithms were used for adaptation decision processes. It will therefore be worth evaluating whether the correct balance between controllability and unobtrusiveness has been implemented in A-POInter.

6.2.2 Evaluating Mobile Usability

Additional problems are introduced when evaluating mobile applications over desktop applications. Mobile environments differ from fixed indoor environments such as offices in several ways. Internal factors like tasks and goals are different and external factors such as social resources are unpredictable. Aspects of mobile contexts are affected by situations in everyday life. Unplanned context changes will lead to unplanned interactions with the application (Tamminen, Oulasvirta, Toiskallio and Kankainen 2004).

The challenges associated with designing a positive mobile user experience and the problems encountered whilst conducting user experience research have been discussed by several authors (Riegelsberger and Nakhimovsky 2008). Data gathering methodologies can include log file analysis, laboratory-based usability studies, diary studies and observational and ethnographic research (Kjeldskov, B., Als and Høegh 2004; Riegelsberger and Nakhimovsky 2008). Each of these evaluation methods, however, has its own drawbacks. Automated interaction logging, questionnaires and interviews are usually used in field studies as these are less intrusive than other techniques such as 'think-aloud' or diary studies and they allow the system to be used in a natural fashion.

Field studies can reveal problems not otherwise identified in laboratory evaluations (Nielsen, Overgaard, Pedersen, Stage and Stenild 2006). To evaluate whether an AUI provides usability benefits, the evaluation must replicate as closely as possible the conditions under which the AUI is to be used in the real world. This includes the experience level of the particpants selected. Some of these conditions may be very difficult to simulate using simple lab testing. An adaptive system needs to be tested in its actual context of use and since the system might need to learn preferences to detect trends and patterns, this will need to be evaluated over an extended period of time (Meyer *et al.* 1993). Some usability issues have a high chance of going unnoticed in a laboratory study, or they might be noticed yet deprioritised compared to other more important objectives. Research has shown, however, that after using an application for a long period of time, participants can be very vocal about 'small' usability issues that become irritating due to repetition over time (Riegelsberger and Nakhimovsky 2008). The evaluation of A-POInter should therefore be conducted outside of a lab environment, be held over a long period of time and replicate the intended real-world conditions (e.g. with actual users) as closely as possible.

6.2.3 Evaluation Instruments

Many usability questionnaires and scales exist for evaluating software applications such as Software Usability Measurement Inventory (SUMI) (UCC 2007), Questionnaire for User Interaction Satisfaction (QUIS) (HCIL 1998) and Post-Study System Usability Questionnaire (PSSUQ) (Lewis 2002). These questionnaires, however, are considered to be too generic for evaluating mobile usability (Ryu and Smith-Jackson 2005). A Mobile Phone Usability Questionnaire (MPUQ) has been developed from the aforementioned questionnaires, comprehensive usability studies and other mobile related sources (Ryu and Smith-Jackson 2006) and specifically evaluates the usability of mobile phones and PDA/handheld PC software products. The MPUQ covers most usability criteria that the aforementioned questionnaires cover and additionally includes criteria that others lack, namely mental effort, flexibility, pleasurability, task performance and feedback (Figure 6.2). Several usability dimensions are covered by the MPUQ (Figure 6.2). Almost all the subjective usability criteria used are applicable to evaluating adaptive mobile interfaces, however, those most relevant include *effectiveness*, *efficiency* and *satisfaction* (Ryu and Smith-Jackson 2006).

Usability Criteria	SUMI	QUIS	PSSUQ	MPUQ
Satisfaction		•		•
Affect	•		•	•
Mental effort				•
Frustration			•	•
Perceived usefulness			•	•
Flexibility				•
Ease of use	•		•	•
Learnability	•	•	•	•
Controllability	•			•
Task accomplishment	•		•	•
Temporal efficiency	•		•	•
Helpfulness	•			•
Compatibility	•			•
Accuracy				
Clarity of presentation		•		•
Understandability	•	•	•	•
Installation		•		•
Documentation	•			•
Pleasurability				•
Specific Tasks				•
Feedback		•		•

Usability Dimensions	Shackel (1991)	Nielsen (1993)	ISO 9241 and 9126 (1998; 2001)	MPUQ
Effectiveness	•		•	•
Learnability	•	•		•
Flexibility	•			•
Attitude	•			•
Memorability		•		
Efficiency		•	•	•
Satisfaction		•	•	•
Errors		•		•
Understandability			•	•
Operability			•	•
Attractiveness			•	•
Pleasurability				•
Minimal Memory Load				•
Attractiveness			•	•

Figure 6.2: Comparison of usability criteria and dimensions with existing usability questionnaires (Ryu and Smith-Jackson 2006)

Several other questionnaires have been used during evaluations of AUIs. A questionnaire used during the evaluation of a feature-rich adaptable interface (McGrenere *et al.* 2007), presented 13 usability measures relevant to adaptive interfaces. These statements were then rated using a 5-point Likert scale. More recently, an adaptive audio-video museum guide was developed in order to determine the acceptance of adaptive multimedia museum guides (Pianesi *et al.* 2009). Participants expressed their agreement/disagreement by means of a 7-point Likert scale to three different types of questionnaire items, including statements, questions and semantic differential items. The 16 questionnaire metrics used were derived from numerous literary sources. A summary of metrics applicable to the evaluation of mobile AUIs was compiled by comparing these metrics to other usability criteria discussed above and is given in Table 6.1.

Usability Metric	Description / Motivation	Used in
Control/ Controllability	A feeling of being in control of a system (e.g. system adaptations) leads to approaching behaviour. The sense of control can affect a user's perceived ease-of-use (Pianesi *et al.* 2009).	McGrenere *et al.* (2007), Pianesi *et al.* (2009), MPUQ, Jameson (2002), Alvarez-Cortes *et al.* (2007)
Ease of use	This metric typically involves asking users to rate how easy or difficult each task was to complete (Tullis and Albert 2008).	McGrenere *et al.* (2007), Pianesi *et al.* (2009), MPUQ
Feedback/ Noticeability	Measuring noticeability involves pointing out a specific aspect of a program and asking participants whether they had noticed that aspect during a task (Tullis and Albert 2008).	Pianesi *et al.* (2009), MPUQ
Learnability	Learnability assess how much time or effort is required to achieve maximum efficiency (Tullis and Albert 2008).	McGrenere *et al.* (2007), MPUQ
Quickness/ Temporal Efficiency	Efficiency is the amount of effort required to complete the task and is usually measured when a user has to complete the same task many times. It is usually measured as task completion per unit of time (Tullis and Albert 2008).	McGrenere *et al.* (2007), MPUQ
Satisfaction/ Frustration	This measures the degree to which a user was frustrated or satisfied with his or her experience while performing a task (Tullis and Albert 2008).	McGrenere *et al.* (2007), MPUQ
Usefulness	It can be valuable to look at the difference between the noticeability or awareness of a specific aspect of a system and the perceived usefulness of the same aspect. If the vast majority of participants are unaware of some functionality, yet find it very useful, one should promote or highlight that functionality in some way (Tullis and Albert 2008).	Pianesi *et al.* (2009), MPUQ

Table 6.1: Summary of usability metrics applicable to the evaluation of mobile AUIs

6.2.4 Proposed Evaluation Method

A controlled laboratory evaluation as well as a field study of POInter revealed that users were highly satisfied with the interaction design and functionality of the system (Hill and Wesson 2008). During development of A-POInter, user testing was conducted throughout design and implementation of the system in order to identify specific usability problems (Chapter 5). As discussed in the previous section, it is important to evaluate a mobile application in its intended context of use, over a longer period of time. The evaluation aimed to identify overall user experience and acceptance of A-POInter over a long period of time. A field study of A-POInter was therefore conducted and is discussed in the following section.

6.3 Field Study

6.3.1 Evaluation Goals

The thesis statement in Section 1.4 stated that an adaptive interface will provide usability benefits for mobile PBS, specifically in the areas of ease of use, satisfaction and usefulness. The goal of the evaluation was therefore to determine how participants felt about using A-POInter. Self-reported data gives the most information about a user's perception of the system and their interaction with it. If participants experience positive emotions when performing a task in a system, they will have a higher level of acceptance of and intention to use the system in future. A participant's subjective reaction to a website, product or store may therefore be the best predictor of their likelihood to return or make a purchase in the future (Tullis and Albert 2008).

To determine whether the research objectives were addressed, an international field study of A-POInter was undertaken and participants' self-reported metrics captured. A brief description of each of the metrics collected is given in the following section.

6.3.2 Metrics to be Measured

Usability metrics to be captured during the evaluation of A-POInter were selected (Section 6.2.3) according to the thesis statement (*"An AUI will provide usability benefits for mobile PBS, specifically in the areas of ease of use, usefulness and satisfaction")*. Metrics selected from Table 6.1 for the evaluation of A-POInter included the following: *controllability*, *ease-of-use*, *feedback/noticeability*, *satisfaction* and *usefulness.* Two metrics, namely *learnability* and *efficiency*, were discarded as these were not regarded as relevant.

Literature emphasises the importance of a user retaining perceived controllability of an adaptive system (Section 6.2). Overall user perceptions of control over adaptations in A-POInter were therefore measured. An additional metric, namely *intention to use* was also measured as part of gauging participants' overall impressions of A-POInter. This metric explicitly measures a participant's intention or desire to use the product in the future and demonstrates user acceptance of a given technology (Tullis and Albert 2008; Pianesi *et al.* 2009).

6.3.3 Task Selection

Participants were provided with a test plan, which contained a list of tasks and general guidelines regarding the functionality of A-POInter that they were required to use during the field study. Participants were advised to run through the test plan at least once, before experimenting freely with the system in a natural manner for the duration of the study.

Tasks included selecting, navigating and manipulating a map, searching for POIs, viewing and filtering search results, interacting with specific POI search results and making use of the various GPS functionalities available. The tasks performed were the same tasks used during the requirements field study (excluding the GPS functionality) to derive the adaptation requirements during the field study discussed in Chapter 4. A copy of the field study test plan can be found in Appendix C.

6.3.4 Instrument and Questionnaire Design

Participants were asked to complete a post-test questionnaire to evaluate the various adaptations provided by A-POInter (Appendix D). The questionnaire was structured similarly to the questionnaire discussed in Chapter 4 so that the results of the evaluation could be easily compared to the requirements identified in the earlier field study. Questions were separated into three sections: Information (Data) Adaptation, Interaction (User Interface) Adaptation and Visualisation (Presentation) Adaptation.

The three adaptation sections of the questionnaire (Sections B-D) were composed of statements scored using a semantic differential-based scale (e.g. 1 = Frustrating, 5 = Satisfying). Section E (General) was constructed using a Likert agreement scale. A section at the end of the questionnaire was provided for participants to indicate which feature of A-POInter they enjoyed the most (positive aspect), which feature they disliked the most (negative aspect) and any other adaptation ideas or suggestions for improvement.

6.3.5 Field Study Procedure

It was important that participants tested A-POInter on their own personal mobile device in order to avoid having to carry around a separate 'research' phone, or avoid any lack of familiarity with another mobile phone's interaction techniques which might negatively impact usability. A 'crowd-sourcing' recruitment strategy was followed, whereby A-POInter (and all accompanying documentation) was uploaded to the Internet and members of the public invited to download and test the system using their own personal mobile device. Ethics approval was obtained from the NMMU REC-H Committee. The selection of participants is discussed in Section 6.3.6.

A pack of pre-downloaded map images around Port Elizabeth and the Eastern Cape was available for download, to reduce the need to use a data connection whilst testing A-POInter. Most participants (especially those whose devices featured external memory cards) chose to download and install the map pack. International participants wishing to test A-POInter's GPS functionality, were required to download map data for their area using their device's Internet connection (e.g. WiFi or 3G).

Participants were instructed to use A-POInter for a minimum period of a week, testing A-POInter on at least five days during that week for a period of at least 10 minutes at a time. Participants were encouraged to use A-POInter in as natural fashion as possible, exploring maps and searching for POIs.

6.3.6 Selection of Participants

The majority of international participants were recruited from Mob4Hire, the leader in real world, crowd-sourced, mobile application testing (Mob4Hire 2009). Mob4Hire connects developers with crowd sourced testers that have registered on the website, to bid on submitted projects. Other Web 2.0 social networking systems were used to recruit participants, including advertising the project on several Facebook Windows Mobile "Fan Pages" or "Groups" and Windows Mobile related community forums such as Microsoft Developer Network (MSDN). Additionally, an advertisement was placed on the homepage of the Nelson Mandela Bay Tourism website (NMBT 2009). A total of 29 people confirmed their initial participation in the study. Unfortunately, nearly half of these recruits were dropped from the study for several reasons, which included a stolen phone, misplaced phone, broken screen, and failure to successfully run A-POInter (e.g. due to service provider restrictions). Additionally, several recruits ceased all communication with the author after signing up for the test and therefore failed to return the questionnaire.

A total of 15 participants successfully completed the study and returned the questionnaire. Nine international participants were recruited through Mob4Hire and had bids accepted ranging from $5 US to $20 US. The remaining six local participants were offered R50 boardwalk shopping vouchers for completing the test. The demographics (Figure 6.3) are summarised as follows: 80% were below the age of 39; 87% were male; six (40%) were local participants and nine (64%) were international participants spanning four continents. The majority of participants (93%) had over five years of general computer experience; and the majority (80%) had experience with a smartphone or PDA (at least one year).

The age group of participants was comparable to those in the requirements analysis field study. Sixty percent of participants were below the age of 39. There were far more male participants (87%) compared to the prior field study (50%). The spread between local and international participants was similar in both field studies. Both field studies included more international participants than local participants. The majority of international participants in both studies originated from Europe. Both field studies featured participants with a high level of computer experience, however the evaluation field study featured participants who were experienced with smartphones/PDAs unlike the earlier field study. The high smartphone/PDA experience levels ensured that usability results were not skewed due to lack of familiarity with the application platform.

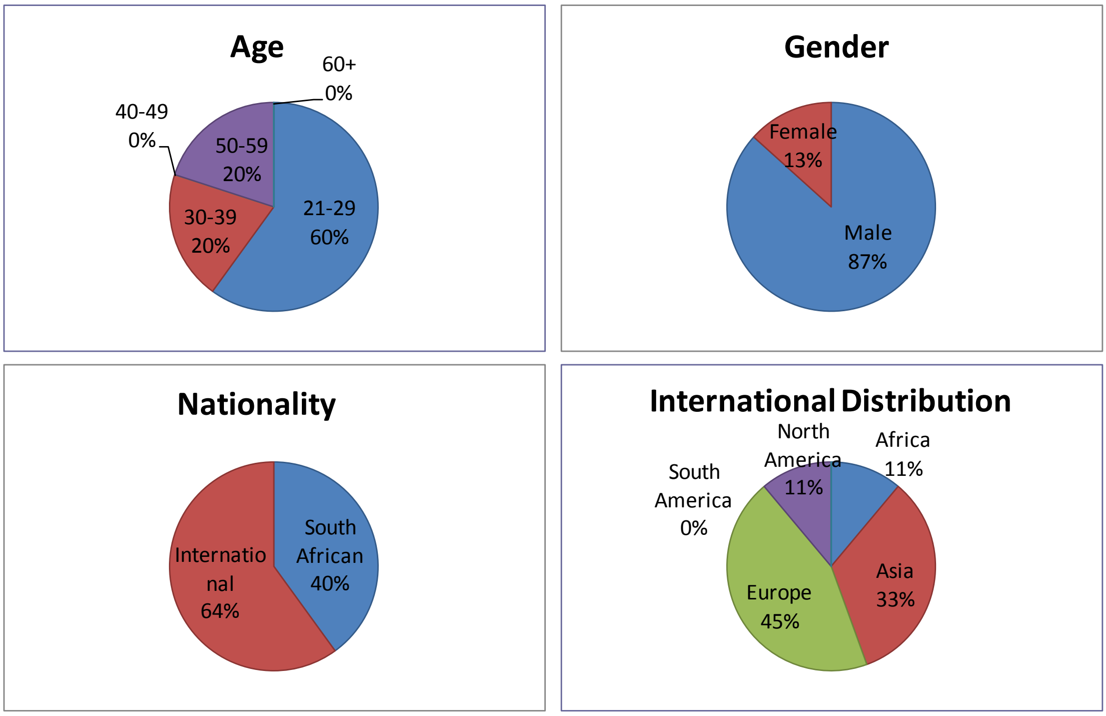


Figure 6.3: Participant demographics (n=15)

6.4 Analysis of Results

Data capture was performed using Microsoft Excel (Appendix E). Sections 6.4.1-6.4.3 contain the results of the questionnaire. A measure of central tendency for each response was calculated using descriptive statistics (arithmetic mean with standard deviation, median and mode), using the same scale as in the requirements analysis field study (Chapter 4). A median score of 4 or higher was considered favourable, 2 or below as unfavourable and 3 as neutral or indecisive. For median scores of 3, the mean was examined. If the mean was less than 2.6 it was considered unfavourable, if the mean was above 3.4 it was considered favourable. The derivation of the interpretation intervals was explained in Section 4.4.

6.4.1 Information Adaptation

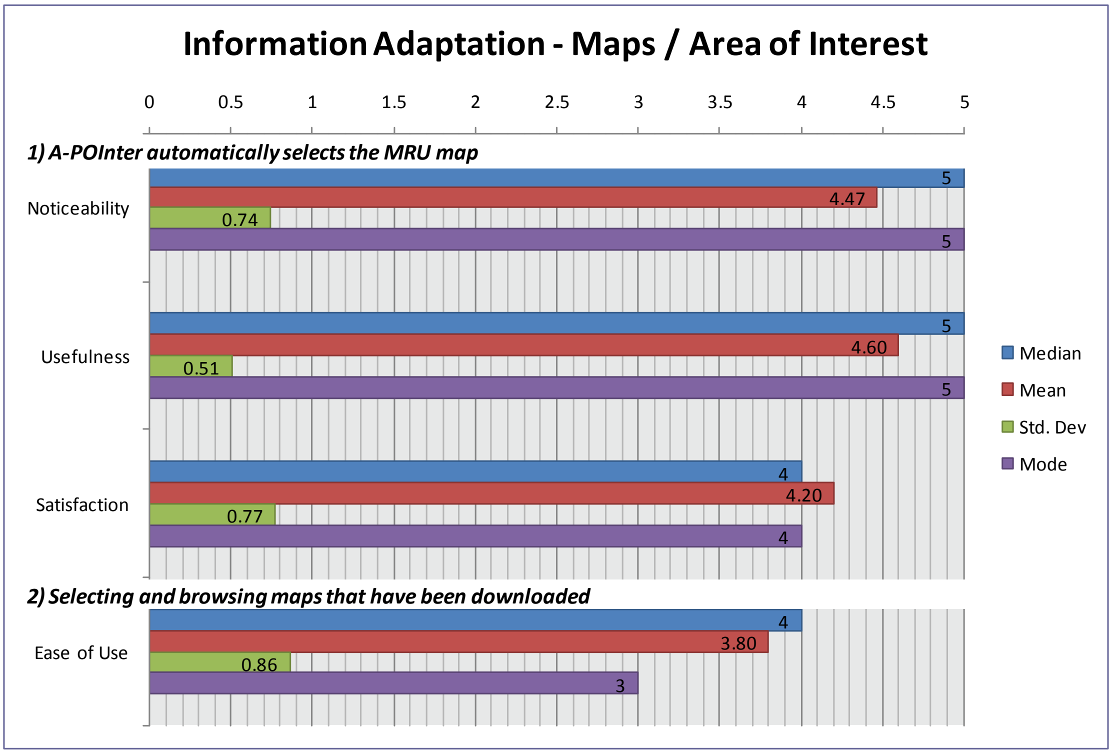


Figure 6.4: Information Adaptation - Maps / Area of Interest (Scale Ratings) (n=15)

The results of Section B1-B3 (Maps / Area of Interest) are summarised in Figure 6.4. Participants found it highly noticeable (median = 5) and highly useful (median = 5) that A-POInter automatically selected the MRU map. Participants were generally satisfied with this adaptation feature (median = 4).

Participants found selecting and browsing maps that had been downloaded to be relatively easy (median = 4). A high standard deviation was obtained for this item (0.86). Variations in ratings for this questionnaire item might be attributed to the speed at which new maps are downloaded and cached to the mobile device, which is dependent on the type of Internet connectivity used such as GPRS/3G as well as signal strength. A couple of participants reported that the Hybrid map-style was not always available for them and that they had to download and use the Road map-style. It was established that the area in which these participants were browsing (typically their home neighbourhood) did not have detailed satellite map imagery available at the zoom level that they were attempting to view and that this was therefore not an issue with A-POInter itself.

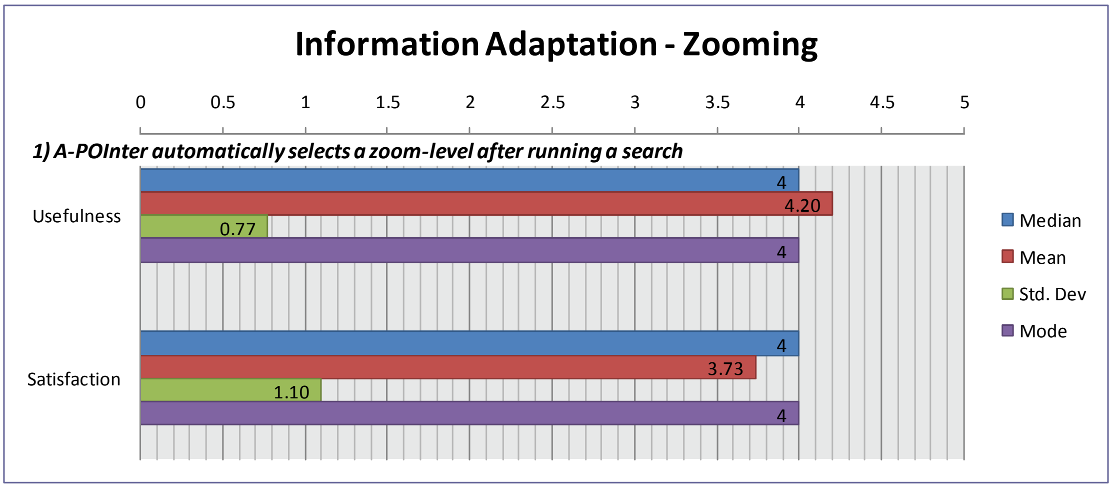


Figure 6.5: Information Adaptation – Zooming (n=15)

Figure 6.5 summarises the results of Section B4 (Zooming). Participants found automatic zooming useful (median = 4) and were generally satisfied (median = 4) when A-POInter automatically selected a zoom-level after running a search for POIs. A high standard deviation was, however, recorded for satisfaction (stdev = 1.10). Some participants explicitly expressed that they liked it when A-POInter automatically adjusted the zoom-level, especially once map images for that area had been cached. One participant found it confusing when he zoomed into a "random town", ran a search for POIs (in that area), only to have A-POInter zoom out from that town and zoom back in on Port Elizabeth. This occurred because there are a limited number of POIs in A-POInter (for testing purposes) and they are all situated in and around Port Elizabeth. A-POInter therefore panned and zoomed to the nearest POI search result (as described in Section 5.4.1.5).

The results of Section B5-B7 (Searching and POI Search Results) are summarised in Figure 6.6. Very high ratings were recorded for both the noticeability and usefulness of A-POInter automatically selecting a user's preferred search criteria (median = 5). Participants were also generally satisfied with this adaptation (median = 4).

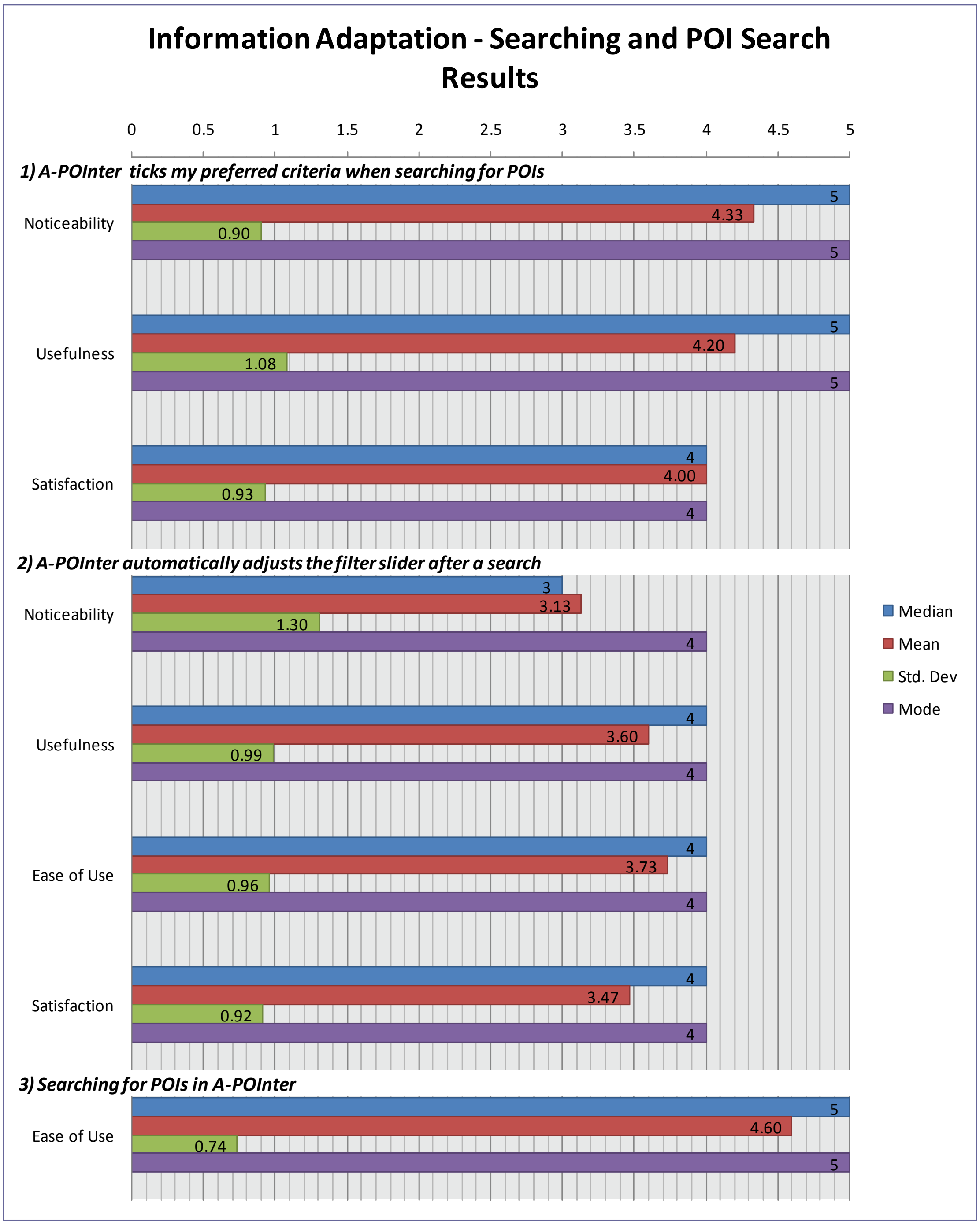


Figure 6.6: Information Adaptation - Searching and POI Search Results (n=15)

Few participants found it noticeable that A-POInter automatically adjusted the filter slider after running a search (mean = 3.13). Participants who were not aware of this adaptation feature, learned of its existence either via the post-test questionnaire, or through post-test discussion with the author via electronic communication. This indicates that it might be useful to provide visual feedback to the user (e.g. highlight or animate the filter percentage label after it has been automatically adjusted). Participants agreed, however, that this was a useful feature (median = 4), were satisfied with the adaptation (median = 4) and found it easy to use. One participant explicitly stated that she would have preferred the filter percentage to always remain constant and only change when she manually adjusted the slider.

A high rating was recorded for the ease with which participants could search for POIs in A-POInter. Positive qualitative comments related to searching for POIs included the following:

- *"The most positive aspect is that one is able to get information of nearby petrol pumps and even search for hotels with ease…"*
- *"I like this program. It's easy to find POIs…"*
- *"Searching for POIs was very an enjoyable feature. To be able to narrow searches based on different criteria was very useful"*
- *"I enjoyed searching POIs based on categories i.e. Mexican restaurants, 3-star hotels etc"*
- *"I have only had a PDA phone since early this year so could still be considered a novice user. I therefore found it useful when A-POInter automatically helped me along with tasks, for example by ticking my preferred criteria when searching".*

Key positive aspects that can be identified from these comments received, related to ease of use, usefulness (including automatic task completion) and enjoyment (satisfaction), which address all three aspects of the thesis statement.

6.4.2 Interaction Adaptation

Figure 6.7 summarises Section C1 (Menu and List Options). Most participants noticed that A-POInter reordered the search criteria by placing preferred criteria at the top of the list and were generally satisfied with this adaptation feature (median = 4).

One participant stated the following regarding criteria reordering: *"After repeated use, I actually enjoyed A-POInter selecting and moving my preferred search criteria to the top. This prevented me from having to scroll and reselect those criteria each time I* *searched"*. Another participant, however, stated that he would not like A-POInter to reorder the criteria lists. He stated that: *"this way, when I am in a tearing hurry, I would not need to read all the options again and again. Knowing the order helps to quickly go to that particular part intuitively"*. As most other participants found this feature highly useful and easy to use (median = 5), this particular participant's opinion can be considered an outlier. It does, however, indicate that it might be useful to provide a manual override for this adaptation, allowing a user to disable list adaptation if he/she wishes. This issue relates to the importance of *controllability* discussed in Section 6.2.

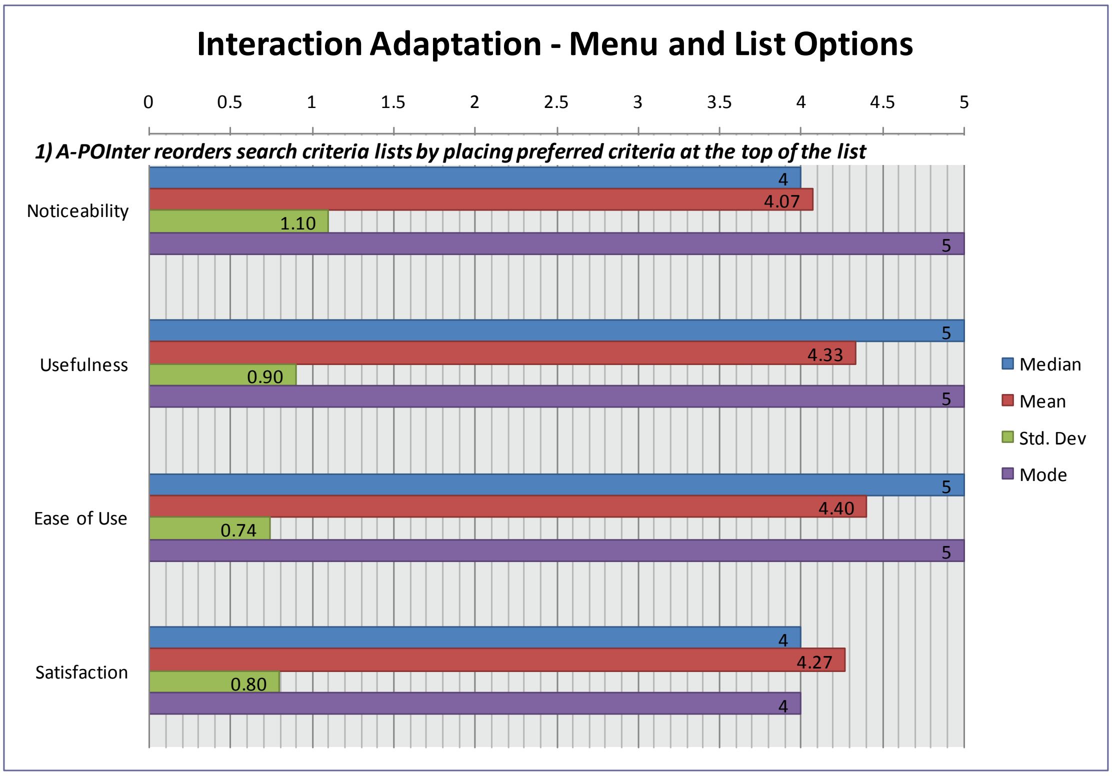


Figure 6.7: Interaction Adaptation – Menu and List Options (n=15)

The results of Section C2-C5 (Zooming and Panning) are summarised in Figure 6.8. Positive ratings (median = 4) were recorded for the noticeability, usefulness and satisfaction of both automatic selection of the preferred zooming and panning techniques. Participants agreed that it was easy to zoom in and out of maps (median = 4) and pan maps in A-POInter (median = 4).

One participant stated that he did not enjoy any of the panning techniques in A-POInter at all. This negative result might be attributed to the slow response time of the system when panning to new map areas (i.e. downloading new maps), when using a slow Internet connection. Alternatively, this might indicate a need to investigate alternative panning techniques, for example using tilt-sensors available in modern mobile devices.

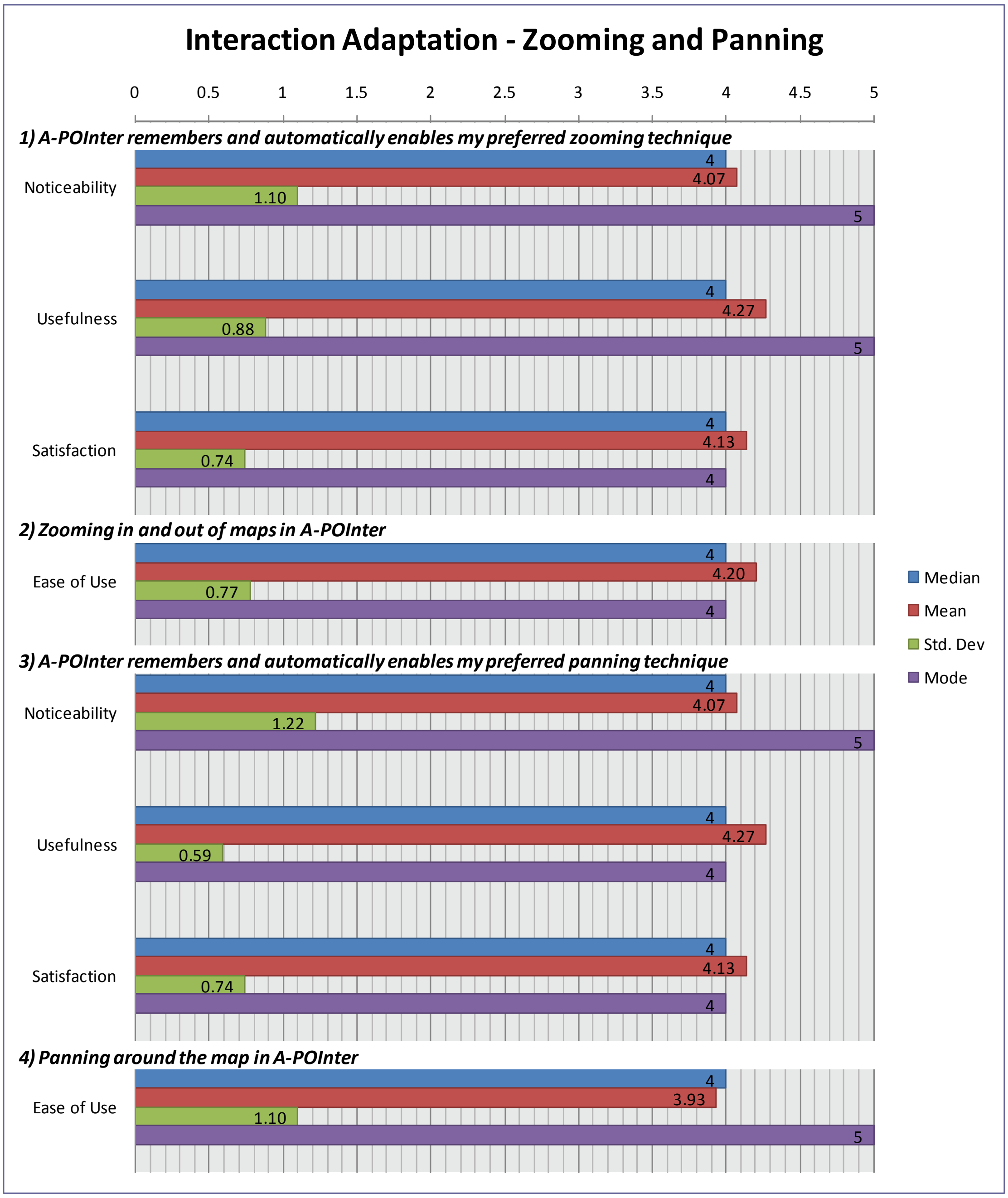


Figure 6.8: Interaction Adaptation – Zooming and Panning (n=15)

One participant suggested that it was not necessary to have an icon on the toolbar for switching the zooming and panning techniques, as users will generally settle on one preferred technique and not need to switch techniques frequently. Removing these two icons would free up space on the toolbar, possibly improving the usability. The user would still be able to switch techniques by accessing the relevant menu option.

A few participants explicitly expressed a high level of satisfaction with the new zooming technique introduced in A-POInter, namely the Bounding-Box technique. It was, however, suggested that it could be further improved by allowing it to "skip" zoom levels (i.e. by drawing a smaller box), allowing them to zoom in faster.

6.4.3 Visualisation Adaptation

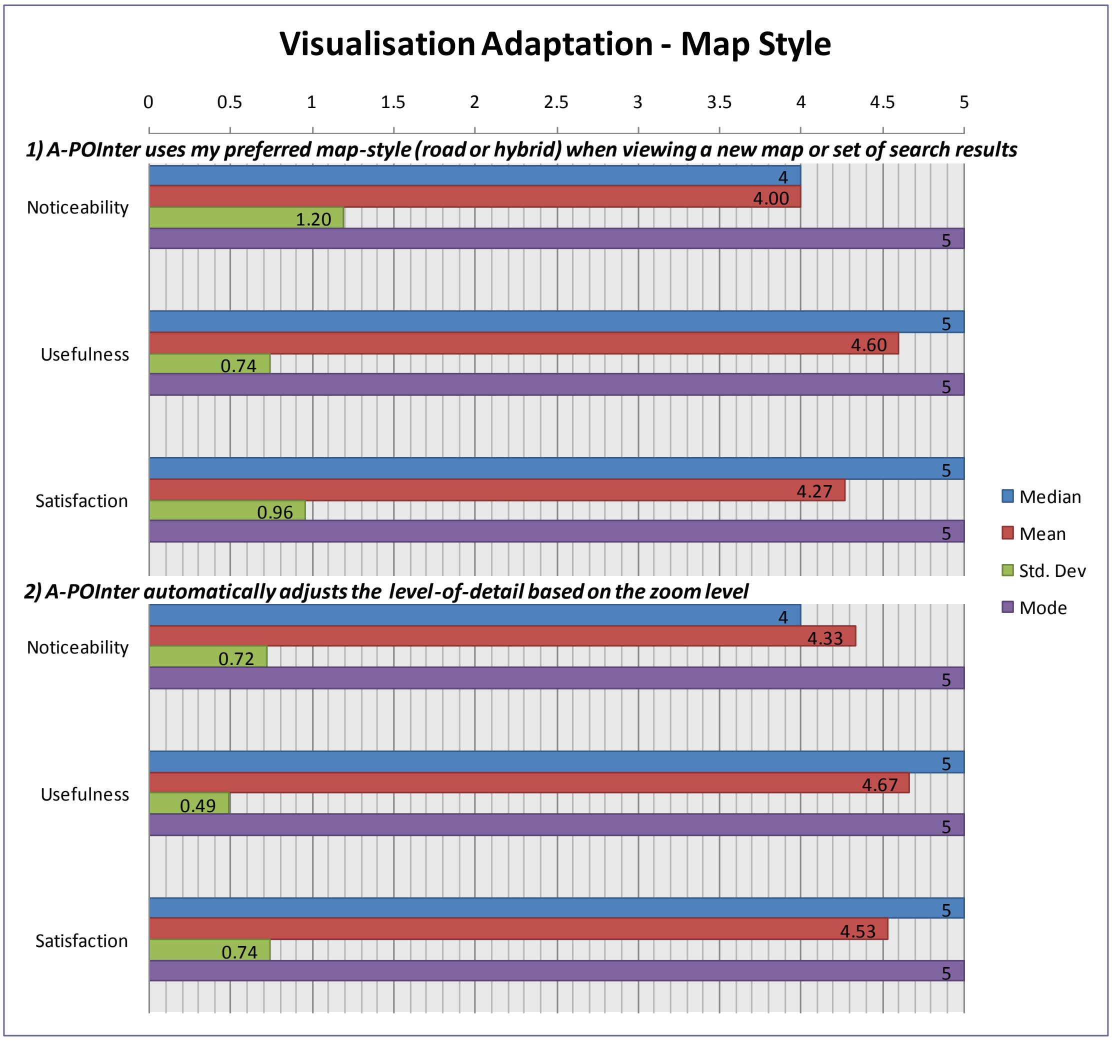


Figure 6.9: Visualisation Adaptation – Map Style (n=15)

Figure 6.9 summarises Section D1-D2 (Map Style). Participants noticed that A-POInter selected the preferred map-style (median = 4) and found it highly useful and satisfying (median = 5).

Most participants noticed that A-POInter automatically adjusted the level-of-detail based on the zoom level (median = 4). Very high ratings (median = 5) were recorded for usefulness and satisfaction regarding this adaptation feature. Regarding the POI thumbnail image available via the "View POI details screen", one participant said the following: *"It was good that there is a picture of the POI included, as it adds to the ease of searching POIs, because when actually going to that POI the user will have a* *visual reference to that place"* (Figure 5.11). This suggests that it is more useful to display the exterior of a restaurant or hotel than the interior for thumbnail pictures.

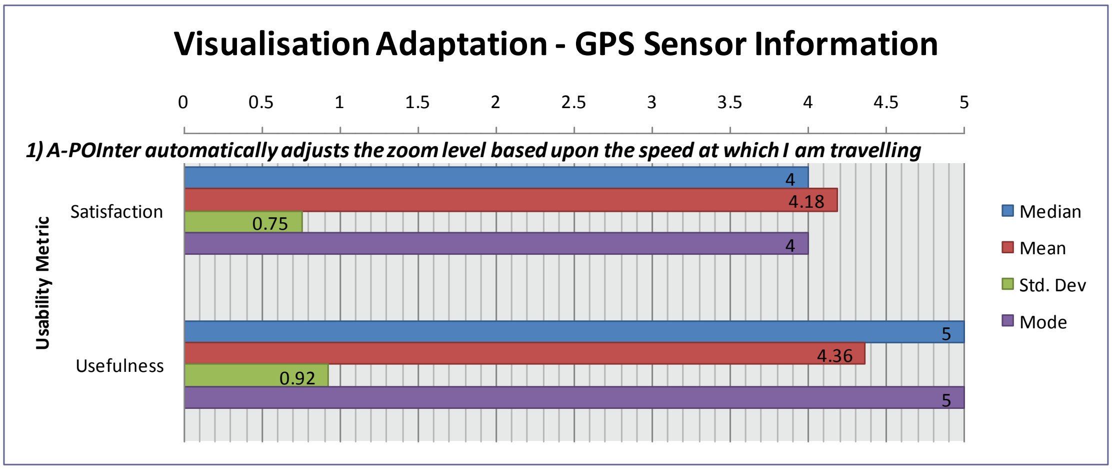


Figure 6.10: Visualisation Adaptation – GPS Sensor Information (n=11)

The results of Section D3 (GPS Sensor Information) are summarised in Figure 6.10. Of the 15 participants tested, 11 had GPS compatible devices and successfully tested the GPS adaptations available in A-POInter. A high level of satisfaction was recorded (median = 4) when A-POInter automatically adjusted the zoom level and panned the map to track the user's location. A high usefulness rating was also recorded (median = 5).

Several participants described the GPS adaptation functionality as the most positive aspect of A-POInter. Qualitative comments included:

- *"I like GPS mode"*
- *"Using the GPS to track my location was the most positive aspect of A-POInter"*
- *"Enjoyed GPS feature / support"*
- *"Being able to locate POIs as I drive"*

A couple of participants suggested additional functionality based on a 'compass' or direction. Qualitative suggestions included:

- *"A direction comparer (with the map and my location) would be very helpful"*
- *"It might be nice to rotate the map to face up in the direction I am travelling, or alternatively to use a directional arrow instead of a car to indicate this..."*

It is possible to incorporate this feature during the integration of turn-by-turn navigation (driving directions feedback) using either the GPS and/or using data from sensors such as magnetometers that are being integrated into new mobile devices.

One participant expressed a desire for A-POInter to provide location-based services, based on GPS data. Qualitative suggestions included:

- *"I suggest that information adapt to suggest criteria according to my current GPS location"*

For the purposes of this research, location-based services were not included in the scope, as discussed in Chapter 1.

6.4.4 Overall Impressions

Section E of the questionnaire related to participants' overall impressions and is summarised in Figure 6.11. All the results were positive, especially those for intention-to-use, learnability and satisfaction, which all received median scores of 5. Other usability metrics measured included control and ease-of-use, which received median ratings of 4.

Several positive qualitative comments were received relating to overall impressions. Comments relating to ease-of-use included the following:

- *"Most positive aspect of A-POInter is that its small and fast for installation. It's very easy to use"*
- *"I like this program, it's easy to find POIs. I like GPS mode."*
- *"Useful and easy to use application while travelling to a country that never had been before. Supplying information about the most significant needs a man would when travelling to foreign countries"*
- *"Simplicity of use"*

Other general comments received included the following. The last two comments refer to the Halo Interface for off-screen POIs (Section 2.4).

- *"This is an amazing application. I have used various such applications including Google Maps and Google Navigation, but this seem to be much more interesting"*
- *"Carry on with the application it is good"*
- *"I liked the red circles around the search locations, this feature helped to move to the locations quickly"*
- *"Most positive aspect is indicator for off-screen POIs"*

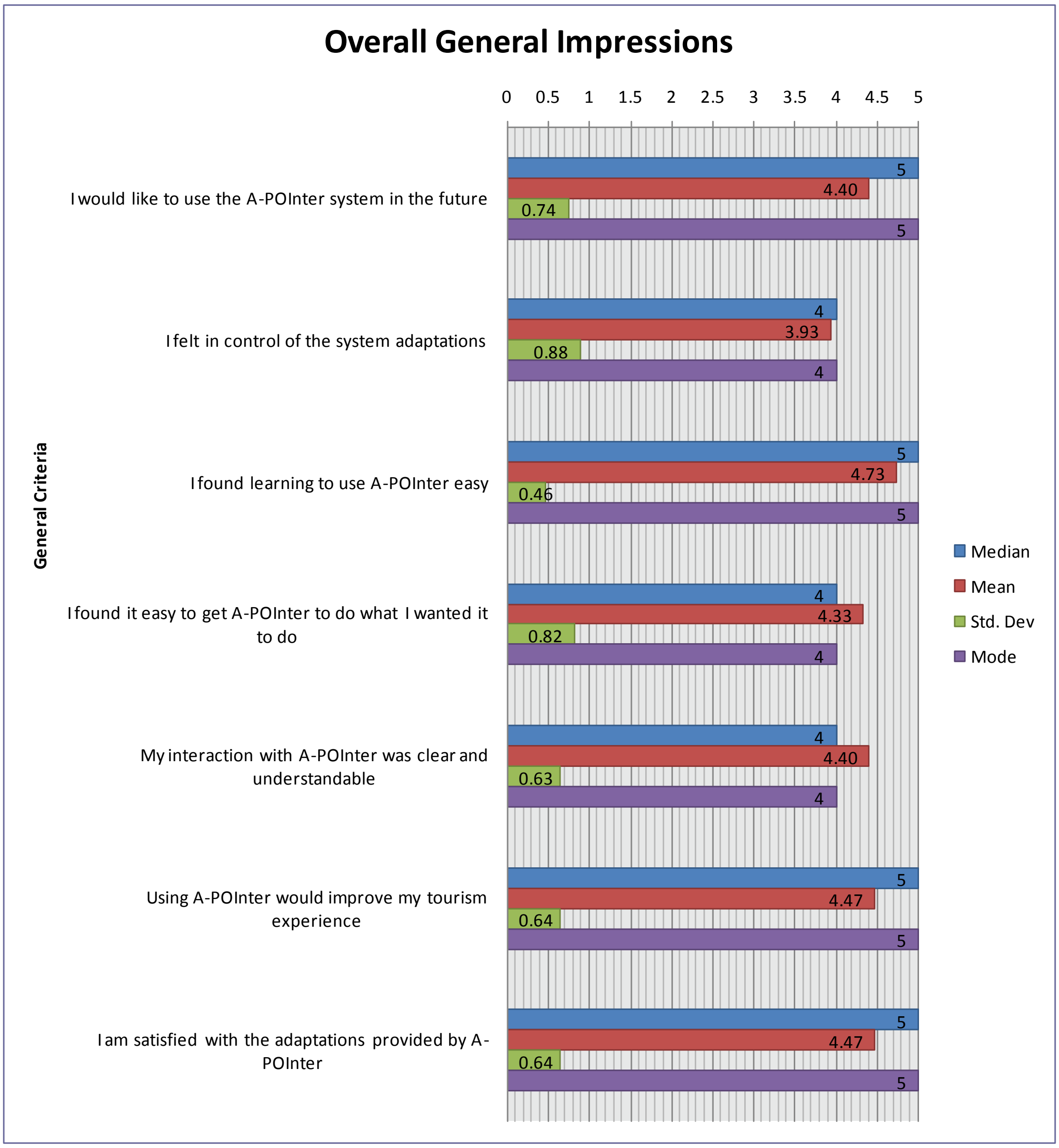


Figure 6.11: Overall General Impressions (n=15)

6.5 Additional Qualitative Feedback

Several other suggestions were received that do not relate specifically to any of the questionnaire sections or were not discussed in earlier sections. These were analysed and sorted into functionality, usability and adaptation suggestions and are discussed below.

a) Advanced Functionality Suggestions

- Add the ability to manually search for a street name / address
- Incorporate turn-by-turn navigation (driving directions)
- Add auto-detection of GPS settings (hardware port number etc) and add a separate satellite monitor window

- Increase number of POIs in database
- Allow users to easily upload or add their own POIs *(due to prototype limitations it was not feasible to substantially expand the number of POIs in the database)*

b) Usability Suggestions

- Remove some superfluous message-box popups (e.g. confirmation of successful deletion after removing a favourite)
- Extend GPS timeout length
- Move manually saved searches from "History" to "Favourites"

c) Adaptation Suggestions

- Allow the user to view and edit ALL system adaptations in a "Settings" type menu or list (i.e. edit the User Model)
- Allow manual override of ALL system adaptations so that any one adaptation can be manually activated/deactivated

6.6 Discussion

The goal of the evaluation of A-POInter was to elicit participants' perceptions when interacting with A-POInter, specifically in the areas of ease of use, satisfaction and usefulness. High ratings for ease-of-use (median >= 4) were obtained for all the key tasks, namely selecting and browsing maps (Figure 6.4), searching for POIs (Figure 6.6) and zooming and panning (Figure 6.8). Participants indicated that they were highly satisfied overall (median = 5) with the adaptations provided by A-POInter (Figure 6.11).

High ratings were received for noticeability and usefulness (median >= 4) for all adaptations in A-POInter, with the exception of the filter adjustment adaptation. A lower noticeability rating for this adaptation feature was, however, mitigated by a high usefulness rating (Figure 6.6). It would therefore be worthwhile to investigate highlighting the filter adaptation when it occurs (e.g. by flashing the filter icon), to increase awareness which can have an effect on perceived control. It was encouraging to note that the other metrics measured for all adaptation features (usefulness, satisfaction and ease-of-use) returned consistently positive results.

Constructive suggestions and qualitative comments were obtained from the participants regarding all aspects of A-POInter. Negative feedback relating to device slowness or map detail was attributed to Internet connectivity issues or lack of map coverage (i.e. beyond A-POInter's control). Several suggestions were given for improving the usability or adding functionality to A-POInter. The only suggestions related to system adaptations were requests for greater control over manual overrides and manually editing user profiles (of learned preferences), both of which relate to controllability.

Adaptation Requirement (Chapter 4)	Usefulness (Median)	Satisfaction (Median)
Information Adaptation		
1) Use the MRU area of interest to suggest a starting map	5	4
2) Automatically select the most appropriate zoom level (in order to contain all relevant POI icons onscreen) after running a search.	4	4
3) Automatically adjust the level of detail for the map based on the current zoom level (e.g. showing specific details such as the POI name at closer zoom levels).	5	5
4) Suggest MFU selections when entering search categories and criteria.	5	4
5) Automatically run a filter to show only the most relevant POI search results according to the criteria specified and their preference history (i.e. A-POInter should take preferred POIs and preferred criteria (MFU) into consideration). A-POInter must provide the ability to adjust the filter to view all search results if desired.	4	4
Interaction Adaptation		
1) Reorder menu items by placing the MFU selections at the top of the list.	5	4
2) *Not* hide any menu items.	n/a	n/a
3) Place both MRU and MFU selections at the top of the list when specifying search criteria.	n/a	n/a
4) Remember the preferred zooming technique (based on MFU) and set it as the default (i.e. other zooming technique(s) are not available unless selected).	4	4
5) Provide a means to quickly view the surrounding map area (*Overview+Detail / Focus+Context*)	n/a	n/a
6) Remember the preferred panning technique (based on MFU) and set it as the default (i.e. other panning technique(s) are not available unless selected).	4	4
Visualisation Adaptation		
1) Always use the MFU map style.	5	5
2) Automatically adjust the zoom level according to the speed at which the user is travelling. A-POInter should zoom out when travelling faster (e.g. by car) and zoom in when travelling slower (e.g. by foot). An option to override this auto-zoom feature should be provided.	5	4
3) Provide a thumbnail image (when available) of the POI at closer zoom levels instead of showing just the standard categorical icon.	n/a	n/a

Table 6.2: Comparison of evaluation field study ratings received, with the adaptation requirements derived from the initial field study (Chapter 4)

The usefulness and satisfaction ratings for the adaptive features in A-POInter were compared to the adaptation requirements derived from the earlier field study (Section 4.5) and are summarised in Table 6.2. Consistently high ratings (median >= 4) were received for usefulness and satisfaction for all the adaptation requirements implemented. The results obtained therefore confirm the validity of the adaptation requirements identified in Chapter 4 and support the thesis statement.

6.7 Conclusions

Several techniques for evaluating AUIs and mobile applications together with the associated difficulties were investigated to determine the most appropriate evaluation methods for A-POInter. This investigation highlighted the importance of conducting a field study in the system's intended environment and context of use. An international field study involving 15 participants was conducted and post-test questionnaires issued to participants. Participants used A-POInter on their mobile device for a minimum period of a week.

Several usability metrics applicable for evaluating mobile AUIs were identified in Table 6.1. The metrics used to evaluate A-POInter were ease-of-use, satisfaction, usefulness and other self-reported metrics such as noticeability, intention to use and control. These metrics were chosen to enable the research objectives to be evaluated.

The field study results revealed that participants were highly satisfied with A-POInter's adaptations and regarding them as easy to use and useful therefore supporting the thesis statement. It can be concluded therefore, that an adaptive mobile PBST can provide several usability benefits. The consistently positive ratings (median >= 4) for all the adaptation features implemented in A-POInter provides further evidence to support that the adaptation requirements identified in Section 4.6 are valid and should be implemented in an adaptive mobile tourist guide.

The following chapter concludes the dissertation. The research achievements and contributions are summarised, problems encountered and implications for existing and future research discussed.

Chapter 7: Conclusions and Recommendations

7.1 Introduction

This chapter concludes the dissertation. The research objectives are restated and the extent to which these goals were satisfied is discussed. The contribution of this research is summarised. Some problems that were encountered during the research are discussed and opportunities for future research and development are presented.

7.2 Research Objective Achievements

The thesis statement was: "*An AUI will provide usability benefits for mobile PBS, specifically in the area of ease-of-use, usefulness and satisfaction*".

The objectives of this research were as follows:

- To investigate the limitations of existing mobile PBSTs;
- To propose requirements for a mobile PBST;
- To identify the potential benefits of AUIs;
- To investigate how AUIs can be used to support mobile PBS;
- To select an appropriate model to support the development of an AUI for a mobile PBST;
- To derive adaptation requirements for a mobile PBST;
- To develop an AUI, called A-POInter, to support mobile PBS;
- To identify the usability benefits of an adaptive mobile PBST by evaluating the usability of A-POInter and;
- To derive recommendations and conclusions based on the results of this research.

Chapter 2 identified and summarised several limitations of existing mobile PBSTs. These limitations were categorised into information, interaction and visualisation aspects. It was discussed how POInter successfully addressed some of these limitations, however several limitations still remain such as the issue of occlusion. Shortcomings of existing PBS algorithms were identified in Section 2.5. A multi-criteria PBS algorithm was implemented in POInter that built upon the strengths of existing algorithms and addressed some of the shortcomings of existing algorithms. The design of the PBS algorithm used in POInter was investigated and it was motivated that the algorithm was accurate as it demonstrates perfect precision and relevance. Several suggestions to further improve the PBS algorithm used in POInter were identified in Section 2.5.4.5. These suggestions include using fine-tuneable criteria and dynamic weightings to allow hard-constraints to be specified. Improvements to the PBS algorithm were not, however, included in A-POInter as these suggestions were not classified as adaptation and therefore would not support the thesis statement. Chapter 2 concluded with the proposal of requirements for a mobile PBST, based on the review of extant systems and techniques.

The benefits of AUIs were identified in Chapter 3. These benefits include the creation of personalised systems, task allocation or partitioning, reducing information overflow and providing intelligent help to users when using new or complex systems. It was decided that A-POInter would focus on creating a personalised system and reducing information overflow. Several techniques to achieve the benefits of adaptation identified were discussed. These included using adaptive menu systems, Fisheye interfaces, Multiscale or semantic zooming as well as Focus + Context techniques such as the Spiral or SnailList for manipulating long lists. Four main types of adaptation were identified, namely information, interaction, visualisation and technology adaptation. It was motivated that this research should focus on the first three adaptation types. Objects that are potentially adaptable in MMV systems were identified and summarised under the three main adaptation types. This summary was used to select adaptable objects applicable to the development of an adaptive mobile tourist guide, to be called A-POInter.

The benefits of using a model-based approach during development of an adaptive mobile PBST were discussed in Chapter 1. Several models to support adaptation were examined in Section 3.4. An existing model for the design of adaptive MMV systems, called the Proteus Model, was discussed. The Proteus Model addresses the shortcomings of existing models and supports the three main types of adaptation. The Proteus Model was therefore selected as the most appropriate model for the development of A-POInter.

Chapter 4 described a field study undertaken in order to verify a list of provisional adaptation requirements for A-POInter. The adaptation requirements proposed were derived by combining the requirements for a mobile PBST (Chapter 2), with the adaptable objects for a MMV system (Chapter 3). Thirty tourists completed several tasks using POInter after which they completed a post-test questionnaire containing suggestions for adapting the information, interaction and visualisation aspects of the system. The results clearly showed that participants would like to use an adaptive mobile tourist guide. A set of adaptation requirements for a mobile tourist guide was derived from the analysis of the field study and summarised in Section 4.5.

The design and implementation of A-POInter, was discussed in Chapter 5. This included a discussion of how the Proteus Model was used to reengineer POInter to satisfy the adaptation requirements identified in Chapter 4. The implementation tools used, platforms/devices targeted and how they differed to the ones used during development of POInter were discussed. Some adaptations listed in the requirements were not implemented in A-POInter. These included reordering menu items, placing thumbnail images on the map and the implementation of a ‘smooth-zooming’ Focus + Context feature. All other adaptation requirements were successfully implemented.

Existing techniques for evaluating AUIs and mobile applications were discussed in Chapter 6. Appropriate methods and metrics for the evaluation of A-POInter were identified and motivated. An international field study of A-POInter involving 15 participants was conducted in order to identify the benefits of an adaptive mobile PBST. Participants installed and used A-POInter on their own mobile device for a period of a week, after which they completed and returned a post-test questionnaire via email. The questionnaire captured participants’ perceived ease-of-use, satisfaction, usefulness and noticeability of adaptation features, as well as general aspects such as intention to use and controllability. The field study results were highly positive. Participants perceived A-POInter to be easy to use, highly satisfactory and useful therefore supporting the thesis statement. Consistently positive ratings were received for all the adaptation features implemented (Table 6.2). This provided further evidence to support that all of the requirements identified in Section 4.5 were valid and should be implemented in adaptive mobile tourist guides in future.

All of the research objectives were therefore successfully achieved. The following section reflects on the contribution of this research.

7.3 Summary of Research Contribution

Several algorithms to enable the adaptation of A-POInter in the three main areas of adaptation were identified. These included the identification of preferred items based on frequency or recency, threshold value-based algorithms to control visualisation adaptations and a partitioning algorithm that can be used to enable adaptive menu systems. The successful implementation of A-POInter provided evidence to support that these algorithms can be successfully implemented on a mobile platform.

The successful implementation of A-POInter provides evidence that the Proteus Model is flexible enough to support the design of adaptive MMV systems in the future. This also provides substantial evidence that the Proteus Model is suitable for reengineering an existing system to incorporate an AUI. The implementation also served to identify several benefits of using a model-based design approach, including simplicity and efficiency.

The results of the field study evaluation of A-POInter in Chapter 6 were compared to the results of the initial requirements analysis field study in Chapter 4. A consistent and highly positive set of ratings were received for all adaptations implemented. The results showed that incorporating an AUI into an existing system can provide several benefits including ease-of-use, usefulness and satisfaction. This provided substantial evidence not only to support the thesis statement but to validate the set of adaptation requirements for an adaptive mobile tourist guide identified in Chapter 4.

7.4 Problems Encountered

Several problems were encountered during implementation and evaluation of A-POInter. One of the adaptation requirements (methods to quickly view the surrounding map area) could not be implemented due to technical issues. As discussed in Section 5.4.2.3, A-POInter utilises raster images for map tiles and discrete (step-wise) zoom levels. The algorithm required to transition (as smoothly as possible) between two discrete raster map images at differing zoom levels would slow the application down to unacceptable levels, as many other algorithms (e.g. to recalculate search results, display POIs, Halos etc) are being processed simultaneously.

This issue could be solved by using vector map images instead of raster images, however, this would introduce additional issues and disadvantages. The system would no longer exhibit the aesthetic appeal and usefulness of satellite raster images (i.e. hybrid map). The current web service used to provide raster map images (Microsoft Bing Maps) enables A-POInter to be used anywhere in the world. If vector maps were to be used, it is likely that the map coverage would become localised to a specific country, drastically reducing the potential usefulness of the system. As mobile device hardware improves in the future, this adaptation feature may be revisited and added to A-POInter.

Other problems were encountered during the evaluation of A-POInter. Soon after the system was published online, a couple of participants indicated that A-POInter was producing critical errors at seemingly random times. Fortunately it was discovered that changing participant's regional settings to 'English' (instead of 'Russian' for example), instantly solved all program compatibility issues. After this discovery, the installation instructions were updated (in the Task Plan) for future participants, to include the step to check the phone's regional settings.

Initially, close to 30 participants were recruited to test A-POInter. Several test participants failed to complete the test, however, due to technicalities such as a stolen phone, misplaced phone, broken screen and failure to successfully run A-POInter due to service provider restrictions. Several other participants simply failed to return the completed questionnaire in time and/or became 'unresponsive' to all communications transmitted after signing up and accepting the terms of the study.

7.5 Recommendations for Future Research

7.5.1 Functionality

Several potential improvements to the PBS algorithm used in POInter were identified in Chapter 2. These suggestions were not implemented in A-POInter as they did not fall within the project scope as the algorithm is not directly related to adaptation. The algorithm improvements suggested could, however, positively contribute to the quality of search results returned. It would therefore be worthwhile to incorporate the improvements into A-POInter and re-evaluate this aspect of the system.

Several suggestions for the extension of A-POInter were made by participants during the evaluation (Chapter 6). A few suggestions related to improving or adding GPS functionality. Turn-by-turn navigation (driving directions) could be implemented into A-POInter, as well as enabling Location Based Services (LBS) to push information to users depending on their location (such as traffic data, or information on nearby POI promotions). LBS capabilities such as these were not addressed by this research as they did not fall within the project scope. Additional GPS improvements suggested include live satellite monitoring and automatic GPS configuration detection.

Several participants stated that they would like more POIs to be included in the database. Since A-POInter is a prototype, a limited set of POIs (mostly around the Eastern Cape) were included in the system. POIs are currently stored locally on the device (in XML format). In future, a web service could be created, or an existing web service used, to store POIs in an online repository. POIs could then be downloaded by A-POInter and cached to the device as necessary, in a similar fashion to how map images are currently handled. The implementation of a POI web service would also improve the scalability of A-POInter. Users should be allowed to add their own custom POIs to A-POInter and upload them to the web service for access by others.

7.5.2 Adaptation

The importance of user control came through strongly during the evaluation of A-POInter. Perceived control can be improved in A-POInter in several ways. Users should be allowed to edit their own user profile, enabling manual tweaks or corrections to be made to learned preferences, as well as allowing the user to enable/disable adaptive features.

New sensors currently being introduced into state-of-the-art mobile devices can be used to introduce new techniques to manipulate the map in A-POInter. For example tilt-sensors or accelerometers could be used for panning. Multi-touch screens that are able to recognise more than one simultaneous touch point could be used for zooming in or out of the map. Additional interaction techniques such as these are worth implementing and evaluating to determine whether they enhance the usability of adaptive mobile systems.

Multiple user profiles (suggested in the field study discussed in Chapter 4) can be incorporated into A-POInter to differentiate between a user's differing 'Business' or 'Vacation' preferences. Lastly, methods to further reduce occlusion of onscreen objects can be investigated, for example investigating algorithms that use 2D boundary detection to group densely located POIs on the map.

7.5.3 Extended Field Study

It will be worthwhile to evaluate A-POInter on a much larger, long-term scale, to determine if user's perceptions of adaptations change over time. A-POInter has the potential to be an extremely useful tool for both international and local tourists to South Africa. Tourist attracting "big events" such as the 2010 FIFA World Cup, or the Iron Man international triathlon hosted in Port Elizabeth, will attract many tourists to the area. A-POInter could be hosted on the Internet for members of the public to download, or preloaded onto mobile devices for tourists to hire at airport cell-phone stores, as suggested by some participants during the field studies.

REFERENCES

ALVAREZ-CORTES, V., ZAYAS-PEREZ, B.E., ZARATE-SILVA, V.H. and URESTI, J.A.R. (2007): Current Trends in Adaptive User Interfaces: Challenges and Applications. In *Proceedings of Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, Los Alamitos, CA, USA. 312-317, IEEE Computer Society.

AMAZON (2008): *Amazon.com Help: Privacy Notice* [online]. Available at http://www.amazon.com/gp/help/customer/display.html/ref=hp_navbox_lnpriv_note?nodeId=468496 [Accessed on 2009-10-10].

BAUDISCH, P. and ROSENHOLTZ, R. (2003): Halo: a Technique for Visualizing Off-Screen Locations. In *Proceedings of CHI '03: SIGCHI conference on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA. 481-488, ACM Press.April 5-10.

BILLSUS, D., BRUNK, C.A., EVANS, C., GLADISH, B. and PAZZANI, M. (2002): *Adaptive interfaces for ubiquitous web access*. *Communications of the ACM* **45**(5):34-38.

BURIGAT, S., CHITTARO, L. and MARCO, L.D. (2005): Bringing Dynamic Queries to Mobile Devices: a Visual Preference-based Search Tool for Tourist Decision Support. In *Proceedings of Human-Computer Interaction - INTERACT 2005*, Berlin. Lecture Notes in Computer Science, **3585**:213-226, Springer Berlin / Heidelberg.September 28, 2005.

BURKE, R.D., HAMMOND, K.J. and YOUNG, B.C. (1997): *The FindMe Approach to Assisted Browsing*. *IEEE Expert: Intelligent Systems and Their Applications* **12**(4):32-40.

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.

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

DIETERICH, H., MALINOWSKI, U., KÜHME, T. and SCHNEIDER-HUFSCHMIDT, M. (1993): *State of the Art in Adaptive User Interfaces*. In *Adaptive User Interfaces: Principles and Practice*. 13-48. Elsevier Science Inc.

DÖRING, S. (2006): *Cosima T - Preference Based Search Technology in Tourism*. Technical Report Institute of Computer Science, University of Augsburg. http://www.informatik.uni-augsburg.de/lehrstuehle/dbis/db/publications/all_db_tech-reports/tr-2006-24_doe/tr-2006-24_doe.pdf.

DUNLOP, M. and DAVIDSON, N. (2000): Visual Information Seeking on Palmtop Devices. In *Proceedings of HCI2000*, Sunderland, UK. **2**:19-20, Springer.September 2000.

DUNLOP, M., PTASINSKI, P., MORRISON, A., MCCALLUM, S., RISBEY, C. and STEWART, F. (2004): Design and development of Taeneb City Guide - From Paper Maps and Guidebooks to Electronic Guides. In *Proceedings of ENTER 2004: International Conference on Information Technology and Travel & Tourism*, Cairo. 1-10. Springer.

FINDLATER, L. and MCGRENERE, J. (2004): A comparison of static, adaptive, and adaptable menus. In *Proceedings of SIGCHI conference on Human factors in computing systems*, Vienna, Austria. 89-96, ACM.

FINDLATER, L. and MCGRENERE, J. (2008): Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proceedings of CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy. 1247-1256, ACM.

FINDLATER, L., MCGRENERE, J. and MODJESKA, D. (2008): Evaluation of a role-based approach for customizing a complex development environment. In *Proceedings of CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy. 1267-1270, ACM.

FISHKIN, K. and STONE, M.C. (1995): Enhanced Dynamic Queries via Movable Filters. In *Proceedings of Conference on Human Factors in Computing Systems (CHI95)*, Denver, Colorado, United States 415-420, ACM Press/Addison-Wesley Publishing Co.

FRANCE, R.A.R., BERNHARD (2007): Model-driven Development of Complex Software: A Research Roadmap. In *Proceedings of FOSE '07: 2007 Future of Software Engineering*. 37-54, IEEE Computer Society.

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 on Advanced visual interfaces*, Venezia, Italy. 201-208, ACM.

GAJOS, K.Z., WOBBROCK, J.O. and WELD, D.S. (2008): Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy. 1257-1266, ACM.

GARMIN (2009): *Garmin GPS navigation on your mobile phone* [online]. Available at http://www8.garmin.com/mobile/mobilext/ [Accessed on 2009-10-30].

GENA, C. (2005): *Methods and techniques for the evaluation of user-adaptive systems*. *The Knowledge Engineering Review* **20**(1):1-37.

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.

GOOGLE (2009): *Google maps on your mobile phone* [online]. Available at http://www.google.com/mobile/products/maps.html#p=default [Accessed on 2009-11-20].

GROSSMAN, D.A. and FRIEDER, O. (2004): *Information Retrieval - Algorithms and Heuristics*. 2nd Edn, The Information Retrieval Series. Vol. 15. Springer.

HCIL (1998): *QUIS: Questionnaire for User Interface Satisfaction* [online]. Available at http://lap.umd.edu/quis/ [Accessed on 2009-12-07].

HEFLEY, W.E. and MURRAY, D. (1993): Intelligent user interfaces. In *Proceedings of 1st international conference on Intelligent user interfaces*, Orlando, Florida, United States. 3-10, ACM.

HILL, R. (2007): *A Mobile Preference-Based Search Tool for Tourism Decision Support*. Honours Treatise thesis. Computer Science and Information Systems, Nelson Mandela Metropolitan University. Port Elizabeth.

HILL, R. and WESSON, J. (2008): Using Mobile Preference-Based Searching to Improve Tourism Decision Support. In *Proceedings of SAICSIT '08: Annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries*, Wilderness, South Africa. 104-113, ACM.

HILL, R. and WESSON, J. (2009): Designing an Adaptive Mobile Tourist Guide. In *Proceedings of Southern African Telecommunications Networks and Applications Conference (SATNAC 2009)*, Swaziland.30 August - 2 September 2009.

HORNBÆK, K. and HERTZUM, M. (2007): *Untangling the usability of fisheye menus*. *ACM Transactions on Computer-Human Interaction (TOCHI)* **14**(2), Article no 6:1-32.

HUOT, S. and LECOLINET, E. (2006): SpiraList: A compact visualization technique for one-handed interaction with large lists on mobile devices. In *Proceedings of Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, Oslo, Norway. 445-448, ACM.

HUOT, S. and LECOLINET, E. (2007): Focus + Context Visualisation Techniques for Displaying Large Lists with Multiple Points of Interest on Small Tactile Screens. In *Proceedings of Human-Computer Interaction – INTERACT 2007*. Lecture Notes in Computer Science, **4663**:219-233, Springer Berlin / Heidelberg. http://www.springerlink.com/content/aw9061112k42132m/.

IRANI, P., GUTWIN, C., PARTRIDGE, G. and NEZHADASL, M. (2007): Techniques for Interacting with Off-Screen Content. In *Proceedings of INTERACT 2007*. **4663**:234-249, Springer Berlin / Heidelberg.September 07. http://www.springerlink.com/content/j202g44683748810/.

JAMESON, A. (2002): *Adaptive Interfaces and Agents*. In *The Human-Computer interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. 305-330. Mahwah, NJ, USA, Lawrence Erlbaum Associates, Inc.

KIEßLING, W. and KÖSTLER, G. (2001): *Preference SQL - Design, Implementation, Experiences*. Technical Report Institute of Computer Science, University of Augsburg. http://www.informatik.uni-augsburg.de/de/lehrstuehle/dbis/db/publications/all_db_tech-reports/tr-2001-7_kie_koe/tr-2001-7_kie_koe.pdf.

KJELDSKOV, J., B., S.M., ALS, B.S. and HØEGH, R.T. (2004): Is it worth the hassle? Exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Proceedings of Mobile Human-Computer Interaction – MobileHCI 2004*, Glasgow. **3160**:61-73, Springer, Berlin.13-16 September 2004. http://www.springerlink.com/content/lb9vp65ftt4a686v.

LAPPING, A. (2004): Model driven development with Ada. In *Proceedings of SIGAda '04: Proceedings of the 2004 annual ACM SIGAda international conference on Ada*, Atlanta, Georgia, USA. 19-22, ACM.

LEWIS, J.R. (2002): *Psychometric evaluation of the PSSUQ using data from five years of usability studies. International Journal of Human-Computer Interaction* **14**(3&4):463-488.

MACMANUS, R. (2009): *A Guide to Recommender Systems*, ReadWriteWeb. http://www.readwriteweb.com/archives/recommender_systems.php.

MASTHOFF, J. (2003): *The evaluation of adaptive systems*. In *Adaptive evolutionary information systems*. 329-347. Hershey, PA, USA, IGI Publishing.

MCGRENERE, J., BAECKER, R.M. and BOOTH, K.S. (2007): *A field evaluation of an adaptable two-interface design for feature-rich software*. *ACM Trans. Comput.-Hum. Interact.* **14**(1), Article no 3:1-43.

MCNALLY, J. (2002): *Information Architecture for the World Wide Web, Second Edition (Book Review)* [online]. Available at http://www.digital-web.com/articles/information_architecture_for_the_world_wide_web/ [Accessed on 19 June 2008].

MEYER, B., YAKEMOVIC, K.C.B. and HARRIS, M. (1993): Issues in practical application of an adaptive interface. In *Proceedings of 1st international conference on Intelligent user interfaces*, Orlando, Florida, United States. 251-254, ACM.

MICHELIN (2006): *The MICHELIN Guide for PDAs* [online]. Available at http://www.viamichelin.com/viamichelin/gbr/tpl/psg/produits/htm/pda_guide_michelin.htm [Accessed on 3 March 2007].

MICROSOFT (2009): *Bing Maps* [online]. Available at http://www.bing.com/maps/ [Accessed on 01 October 2009].

MIHALCEA, R., MOONEY, R., GHOSH, J. and LEE, D. (2008): *Information Retrieval and Web Search (PowerPoint)* [online]. Available at www.cs.unt.edu/~rada/CSCE5200/Lectures/IREvaluation.ppt [Accessed on 19 June 2008].

MIO (2009): *Mio Technology South Africa* [online]. Available at http://www.mio.com/za/gps-navigation-products.htm [Accessed on 2009-10-30].

MITROVIC, N., ROYO, J.A. and MENA, E. (2005): Adaptive User Interfaces Based on Mobile Agents: Monitoring the Behavior of Users in a Wireless Environment. In *Proceedings of Simposium on Ubiquitous Computation and Ambient Intelligence*. 371-378, Thomson-Paraninfo.September. http://citeseer.ist.psu.edu/mitrovic05adaptive.html.

MOB4HIRE (2009): *Crowdsourced Mobile Application Testing* [online]. Available at http://www.mob4hire.com/ [Accessed on 17 November 2009].

NESTOR, D., O'MALLEY, L., HEALY, P., QUIGLEY, A. and THIEL, S. (2007): Visualisation techniques to support derivation tasks in software product line development. In *Proceedings of 2007 conference of the center for advanced studies on Collaborative research*, Richmond Hill, Ontario, Canada. 315-325, ACM.

NIELSEN, C.M., OVERGAARD, M., PEDERSEN, M.B., STAGE, J. and STENILD, S. (2006): It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *Proceedings of 4th Nordic conference on Human-computer interaction: changing roles (NordiCHI)*, Oslo, Norway. **189**:272-280.14-18 October 2006.

NILLSON, E.G. (2009): *Design patterns for user interface for mobile applications*. *Advances in Engineering Software* **40**(12):1318-1328.

NILSSON, E.G., FLOCH, J., HALLSTEINSEN, S. and STAV, E. (2006): *Model-based user interface adaptation Computers & Graphics* **30**(5):692-701.October 2006.

NMBT (2009): *Nelson Mandela Bay Tourism* [online]. Available at http://www.nmbt.co.za/ [Accessed on 01 October 2009]

PIANESI, F., GRAZIOLA, I., ZANCANARO, M. and GOREN-BAR, D. (2009): *The motivational and control structure underlying the acceptance of adaptive* *museum guides – An empirical study*. *Interacting with Computers* **21**(3):186-200. July 2009.

PREECE, J., ROGERS, Y. and SHARP, H. (2007): *Interaction Design Beyond Human-Computer Interaction*. Chichester, 2nd Edn, John Wiley & Sons Ltd.

REICHENBACHER, T. (2003): Adaptive Methods for Mobile Cartography. In *Proceedings of 21st International Cartographic Conference*, Durban, South Africa. 1311-1322, Document Transformation Technologies.10-16 August 2003. 129.187.175.5/lfkwebsite/fileadmin/user_upload/publications/reichenbacher/ICC2003_Paper.pdf.

REICHENBACHER, T. (2004): *Adaptive Visualisation of Geographic Information on Mobile Devices* Department of Cartography, der Technischen Universitat Munchen. Munich. http://tumb1.biblio.tu-muenchen.de/publ/diss/bv/2004/reichenbacher.pdf.

RIEGELSBERGER, J. and NAKHIMOVSKY, Y. (2008): Seeing the bigger picture: a multi-method field trial of google maps for mobile. In *Proceedings of CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy. 2221-2228, ACM.

RYU, Y.S. and SMITH-JACKSON, T.L. (2005): Development of Usability Questionnaire Items for Mobile Products and Content Validity. In *Proceedings of HCI International*, Las Vegas, Nevada, USA. Lawrence Erlbaum Associates, Inc.22-27 July 2005. http://uweb.txstate.edu/~yr12/Papers/HCII2005_Submission_Items_Final.pdf

RYU, Y.S. and SMITH-JACKSON, T.L. (2006): *Reliability and Validity of the Mobile Phone Usability Questionnaire (MPUQ)*. *Journal of Usability Studies (JUS)* **2**(1):39-53. November 2006. www.upassoc.org/upa_publications/jus/2006_november/ryu_smith-jackson_mobile_phone_questionnaire.pdf.

SCHIAFFINO, S. and AMANDI, A. (2004): *User-Interface agent interaction personalization issues*. *International Journal of Human-Computer Studies* **60**(1):129-148. January 2004.

SHARIFI, G., DETERS, R., VASSILEVA, J., BULL, S. and RÖBIG, H. (2004): Location-Aware Adaptive Interfaces for Information Access with Handheld Computers. In *Proceedings of Third International Conference, AH*, Eindhoven, The Netherlands. **3137**:328-331, Springer Berlin / Heidelberg.August 23-26.

SHNEIDERMAN, B. and MAES, P. (1997): *Direct manipulation vs. interface agents*. *Interactions* **4**(6):42-61.

SOMMERVILLE, I. (2007): *Software Engineering*. Essex, England, 7th Edn, Addison-Wesley.

SOUTHAFRICA.INFO (2008): *South Africa's telecommunications infrastructure* [online]. Available at http://www.southafrica.info/business/economy/infrastructure/telecoms.htm [Accessed on 2009-11-20].

STUERZLINGER, W., CHAPUIS, O., PHILLIPS, D. and ROUSSEL, N. (2006): User interface facades: towards fully adaptable user interfaces. In *Proceedings of 19th annual ACM symposium on User interface software and technology*, Montreux, Switzerland. 309-318, ACM.

TAMMINEN, S., OULASVIRTA, A., TOISKALLIO, K. and KANKAINEN, A. (2004): *Understanding mobile contexts*. *Personal Ubiquitous Computing*:135--143.

TULLIS, T. and ALBERT, B. (2008): *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Burlington, Morgan Kaufmann Publishers.

UCC (2007): *SUMI Questionnaire* [online]. Available at http://sumi.ucc.ie/latest.html [Accessed on 2009-12-07].

UNIVERSITY OF MICHIGAN DIGITAL LIBRARY PROJECT (1996): *UMDL Glossary - Definition of semantic zooming* [online]. Available at http://www.si.umich.edu/UMDL/glossary.html [Accessed on 28 March 2008].

VAN TONDER, B. and WESSON, J. (2008): Using adaptive interfaces to improve mobile map-based visualisation. In *Proceedings of SAICSIT '08: Annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries*, Wilderness, South Africa. 257-266, ACM.

VIAPPIANI, P., FALTINGS, B. and PU, P. (2006): *Preference-based Search using Example-Critiquing with Suggestions*. *Journal of Artificial Intelligence Research (JAIR)* **27**:465-503.

ZAGAT (2009): *ZAGAT TO GO* [online]. Available at http://mobile.zagat.com/index.php [Accessed on 2009-11-20].

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 ACM symposium on Applied computing*, Dijon, France. 1008-1012, ACM.

Appendix A: POInter Field Study Test Plan

User Evaluation Test Plan for the POInter System

A Mobile Preference-Based Search Tool (PBST) for Tourism Decision Support

Nelson Mandela Metropolitan University
for tomorrow

---

1. Overview:

POInter is a mobile preference-based search tool that enables tourists to identify points-of-interest (POIs), namely accommodation, restaurants, fuel, transport and tourist attractions, most suited to their needs and constraints. It provides interaction methods for supporting incremental searching of POIs and superimposes the search results as graphical icons upon a 2-dimensional map.

2. Brief Tutorial:

After being demonstrated the basic features of the system, please work through the following tasks at a pace you feel comfortable with. If you have any questions or are unsure about how to complete a specific task, please convey this to the evaluator.

3. Tasks to be Performed

3.1 View a local map from the list

- **Return to the "Home Screen"**
  - *Select "Menu > Home Screen" from the main menu.*
- **View a map of Port Elizabeth**
  - *Select "View Map". Select "Specify Area" and then select "South Africa" as a country and "Port Elizabeth" as a city from the drop down list controls.*
- **View the selected map**
  - Tap the "Map" icon.

3.2 Navigate / manipulate the map

- **Experiment with panning**
  - *To pan, drag the stylus pen (slowly) across the map.*
- **Experiment with zooming**
  - *Use the "Zoom In" and "Zoom Out" buttons located on the toolbar (note that it will zoom into the centre of the screen).*
- **Change the map style**
  - *Select "View > Map Style > Hybrid Map" from the main menu to change the map style from "Road" to "Hybrid" (aerial/satellite view with roads).*
- **Use the "quick-zoom" to view the entire city**
  - *Select "View > Zoom to > Level 6 – City" from the main menu to view an overview.*

3.3 Search for a Point of Interest (POI)

- **Perform a search for Accommodation in Port Elizabeth**
    - *Select "Search > Accommodation" from the Main Menu.*
- **Specify the following criteria for Accommodation:**
    - *Subcategory = Bed and Breakfast; Hotel;*
    - *Price Range = RRR – (i.e. Moderate)*
    - *Stars = Tick both 2 ★★ and 3 ★★★ stars*
- **View the search results on the map**
    - *Tap the "View Map" icon* .
- **Run a filter to show only the POIs with an 85% match or higher**
    - *Tap the "Filter" icon and drag the slider slowly upwards until only POIs with a ± 85% match or higher are displayed.*
    - *Note that POIs that match search criteria perfectly (100%) are highlighted in green* .
- **Zoom to "Level 7"**
- **Off-screen POIs**
    - *Off-screen POIs are indicated using the "Halo Technique" (see figure below)*
    - *Pan the map to view an off-screen POI (see 3.2 above)*

3.4 View POI details

- **Pan and zoom towards any POI that has been visualized on the map**
- **View a POI's specific details**
    - *"Double-tap" on a POI to view its specific details.*
    - *Browse all the details using the tabs.*
    - *Note that you can interact with contact details, save as favourite, etc.*
- **Return to the map**
    - *Tap the "Back" icon* .

4 Conclusion:

Thank you for using the POInter system! Feel free to experiment further with the system as there are many features that were not covered. Direct any queries/difficulties you may have to the evaluator.

Following this, please take the time to fill in a **questionnaire** based upon your experiences.

Appendix B: A-POInter Requirements Analysis Questionnaire

Background:

A system called "*POInter*" was developed in 2007 as part of an Honours Treatise entitled: "*A Mobile Preference-Based Search Tool for Tourism Decision Support"* at the Computer Science & Information Systems (CS&IS) department at the Nelson Mandela Metropolitan University (NMMU).

POInter enables users to identify points-of-interest (POIs) such as restaurants or accommodation most suited to their needs and constraints. Search results are superimposed as icons placed upon a map. A green and red bar to the left of a POI icon indicates the percentage match to the user's search criteria.

The aim of this Masters research is to determine if an adaptive user interface can improve the effectiveness and satisfaction of POInter. The aim of this questionnaire is to determine how best POInter should adapt to user behaviour and preferences, in terms of adapting the information, interaction and visualisation.

Instructions:

After completing the tasks using the mobile device, please mark the options in the following questionnaire which most appropriately reflect your impressions about using the POInter system. Space at the end of the questionnaire has been provided for your written comments.

This experiment and questionnaire should take no more than 10-15 minutes to complete.

Contact Details:

Ryan Tebbutt Hill
Masters Student / Lecturer: Department of Computer Science and Information Systems
Nelson Mandela Metropolitan University
South Campus, Summerstrand
Port Elizabeth, South Africa
Office: 09-02-55 (Masters Lab, Embizweni Building)

Email: ryan.hill@nmmu.ac.za
Web: http://www.nmmu.ac.za/ryanhill
Tel: 27(0) 41 504 2094
Cell: 27(0) 82 530 2481

Questionnaire:

A. Biographical Details						
Age:						
< 20	20 - 29	30 - 39	40 - 49	50 - 59	60+	
Nationality:						
South African	International (please specify country)	……………………………………				
Gender:						
Male	Female					
Computer Experience (years):						
< 1	1 - 2	3 - 4	5+			
PDA/Smartphone Experience (years):						
< 1	1 – 2	3 - 4	5+			

B. Information (Data) Adaptation					
Maps / Area of Interest					
1) POInter should remember my *most recently used* (MRU) area of interest to suggest a starting map.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
2) POInter should remember my *most frequently used* (MFU) area of interest to suggest a starting map.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
3) POInter should use the GPS to suggest the initial area of interest (i.e. map).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
Zooming					
4) POInter should automatically suggest the most appropriate zoom level after searching.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
Searching and POI Search Results					
5) POInter should adjust the number of POI search results shown based upon the current zoom level (e.g. show only more relevant POI results when zoomed out).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
6) POInter should automatically display the *level of detail* for the map based on the current zoom level (e.g. show specific POI details such as the name at closer zoom levels).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
7) When entering search criteria, POInter should suggest my MRU categories and criteria.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
8) When entering search criteria, POInter should suggest my MFU categories and criteria.	Strongly Disagree				Strongly Agree
	1	2	3	4	5

9) I would like POI search results to be grouped according to certain criteria.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
10) I would like POI search results to be grouped according to my preference history.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
11) I would like to be able to apply a filter to view the top POIs (e.g. top 3) according to my search criteria and preference history.	Strongly Disagree				Strongly Agree
	1	2	3	4	5

C. Interaction (User Interface) Adaptation					
Menu Options					
1) POInter should reorder menu items based upon my MRU selections.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
2) POInter should reorder menu items based upon my MFU selections.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
3) POInter should hide menu options that I do not use often.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
Searching					
4) When specifying search criteria, POInter should place my MRU criteria selections at the top of the list.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
5) When specifying search criteria, POInter should place my MFU criteria selections at the top of the list.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
Zooming and Panning					
6) POInter should remember my preferred zooming technique (e.g. either draw a box on the map to zoom into, or use step-wise zoom in/out buttons).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
7) POInter should provide a tool to temporarily zoom out of the map in order to view the surrounding area (after which POInter will automatically return to the previous zoom level).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
8) POInter should remember my preferred panning technique (e.g. tap, hold and drag the map, or tap and hold directional arrows (NESW) at the edges of the map.	Strongly Disagree				Strongly Agree
	1	2	3	4	5

D. Visualisation (Presentation) Adaptation					
Map Style					
1) POInter should suggest my MRU map style (Road, vs. Satellite photo, vs. Hybrid).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
2) POInter should suggest my MFU map style (Road, vs. Satellite photo, vs. Hybrid).	Strongly Disagree				Strongly Agree
	1	2	3	4	5
3) POInter should automatically adjust the zoom level according to the speed at which I am travelling.	Strongly Disagree				Strongly Agree
	1	2	3	4	5
Search Results					
4) When zoomed in, POInter should use a small photograph or image (e.g. depicting the actual landmark) for POI search results instead of the standard categorical icons.	Strongly Disagree				Strongly Agree
	1	2	3	4	5

E. General Comments

If you have any other possible adaptation ideas or suggestions for improvement, please describe these below.

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

…………………………………………………………………………………………………..

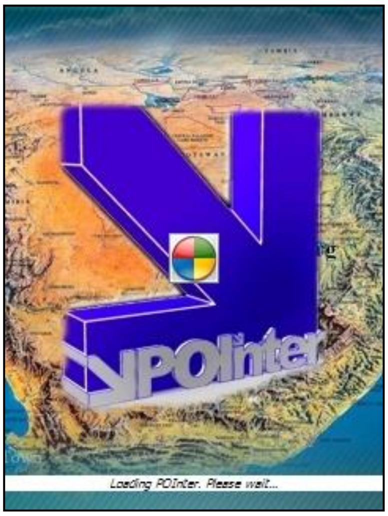
Appendix C: A-POInter Field Study Test Plan

Tasks:

The following tasks are general guidelines regarding the functionality of A-POInter that you should use during the field study and are provided as an example only. It is advisable to run through this test plan in full at least once and thereafter experiment freely with the system for the duration of the field study (for example, search for other POIs such as Restaurants or Tourist Attractions).

1. First time use (Configuring settings and Viewing a Map)

- **Launch A-POInter**
  - After installation (see installation guide), select the "POInter" shortcut from the Windows Mobile start menu.
  - You should be presented with the splash screen (see figure below) while A-POInter is loading.



- **Configure first-time settings**
  - After launching A-POInter and waiting for the program to load (Splash-screen to disappear), you will be presented with a 2x3 grid of icons, known as the "Home Screen".

  - Tap "Settings" found in the bottom left corner. This will bring up the settings configuration page.
  - Select your Internet connection type. This will typically be "Direct connection to the Internet".
  - Tap the "Cache" tab to configure your map cache location and filesize limit. Select "Removable Storage Card" if your device has one, otherwise select "Built-in device memory".

The default cache limit is 50,000 files (approx 500MB). You may change this value if desired. A cache size of at least 1,000 files (approx 10MB) is recommended and higher is better.

- If your device is GPS enabled, specify its program and hardware port as well as baud rate by using the "GPS" tab. These settings should mirror the settings found under: *"Windows Start Menu > Settings > System Tab > External GPS"*.
- Save the settings by tapping the 'Save' icon .

- **View a map of the world**
  - From the Home Screen, select "View Map". Note that "Entire World" is currently selected.
- **View the selected map**
  - Tap the "Map" icon .
- **Exit A-POInter**
  - Select "Menu > Exit POInter"
  - Select "Yes" to confirm the exit.

2. (Re)Launch A-POInter and view a map of Port Elizabeth

- **Launch A-POInter**
  - Select the "POInter" shortcut from the Windows Mobile start menu.
  - After launching A-POInter for the first time, instead of being presented with the Home Screen, A-POInter will automatically take you to your most-recently-used (MRU) map.
- **Change the Map to a map of Port Elizabeth (South Africa)**
  - From now on, to change the map as you did in Step 1 above, select "Menu > Change Map" from the Main Menu.
  - From the "Change Map" screen, Select "Specify Area" and then select "South Africa" as a country and "Port Elizabeth" as a city from the drop down list controls.
  - Tap the "Map" icon .

3. Navigate / manipulate the map

- **Experiment with panning**
  - Select the NESW panning technique from the menu ("View > Panning Technique > NESW Arrows").
  - To pan with this technique, tap or hold any of the arrows at the edge of the map pointing North, West, South or East.
  - Change the panning technique to pen-drag ("View > Panning Technique > Pen Drag").
  - To pan with this technique, drag the stylus pen (slowly) across the map.
  - You can quickly toggle the panning technique by tapping either the or toolbar icons (depending which technique is currently activated).
- **Experiment with zooming**
  - Select the Step Wise panning technique from the menu ("View > Zooming Technique > Step Wise").
  - To zoom using this technique, use the "Zoom In" and "Zoom Out" icons located on the toolbar (note that it will zoom into the centre of the screen).
  - Change the zooming technique to bounding box ("View > Zooming Technique > Bounding Box").

- To zoom in with this technique, tap the bounding box icon located on the toolbar. The icon turns green signifying that you can now draw a box on the map. Draw a box on the screen to zoom into that area.
- You can quickly toggle the zooming technique by tapping either the or toolbar icons (depending which technique is currently activated).

- **Change the map style**
  - Select "View > Map Style > Hybrid Map" from the main menu to change the map style from "Road" to "Hybrid" (aerial/satellite view with roads).
- **Use the "quick-zoom" to view the entire city**
  - Select "View > Zoom to > Level 6 – City" from the main menu to view an overview.

4. Search for a Point of Interest (POI)

- **Perform a search for Accommodation in Port Elizabeth**
  - Select "Search > Accommodation" from the Main Menu.
- **Specify the following criteria for Accommodation:**
  - Subcategory = Bed and Breakfast; Hotel;
  - Price Range = RRR – (i.e. Moderate)
  - Stars = Tick both 2 ★★ and 3 ★★★ stars

- **View the search results on the map**
  - Tap the "View Map" icon.
  - A-POInter will automatically select the most appropriate zoom level, filter level and pan to the nearest POI.
- **Manually change the filter to show POIs with a 75% match or higher**
  - Tap the "Filter" icon and drag the slider slowly upwards until only POIs with a ± 75% match or higher are displayed.
  - Note that POIs that match search criteria perfectly (100%) are highlighted in green.
- **Zoom to "Level 7"**
- **Off-screen POIs**
  - Off-screen POIs are indicated using the "Halo Technique" (see figure above)
  - Pan the map to view an off-screen POI

5. View POI details

- **Pan and zoom towards any POI that has been visualized on the map**
  - Zoom to level 9 above any POI.
- **View a POI's specific details**
  - "Double-tap" on a POI to view its specific contact details, thumbnail image etc.
  - Browse all the details using the tabs (e.g. Classification, Location, Contact).
  - Note that you can interact with contact details (e.g. tap the website link), save the POI as a favourite, etc.
- **Return to the map**
  - Tap the "Back" icon.

6. GPS tracking (if GPS enabled device)*

- **Activate the GPS tracking**
  - Tap the car icon  on the toolbar to activate the GPS
  - **Note: your device must have a GPS built-in and be properly configured under settings (“Menu > Settings > GPS”). If no GPS signal is found after 1 minute, GPS tracking will be disabled.*
  - Once your GPS position has been located, A-POInter will automatically pan the map to your location, place an icon of a car  on the map to signify your position and automatically adjust the zoom level of the map according to the speed at which you are travelling. The map is redrawn every 3 seconds while the GPS is on.
- **Disable the automatic zooming**
  - Some users may want to manually override the zoom level while the GPS is on. To do this, select “View > GPS Tracking > Auto Zoom” (ensure it is unticked).
- **Turn off the GPS tracking**
  - To turn off the GPS tracking, tap the GPS icon  or select “View > GPS Tracking > Track Location”).

Appendix D: A-POInter Evaluation Questionnaire

A-POInter: Field Study Instructions and Questionnaire

Overview of the System:

A-POInter enables users to identify points-of-interest (POIs) such as restaurants or accommodation most suited to their needs and constraints. Search results are superimposed as icons placed upon a map. A green and red bar to the left of a POI icon indicates the percentage match to the user's search criteria.

The aim of this research is to determine the extent to which an adaptive user interface can improve the ease of use, user satisfaction and usefulness of A-POInter. The aim of the questionnaire is to measure the usability (e.g. effectiveness and user-satisfaction) of A-POInter, which includes determining whether the adaptive features implemented provide an effective means of facilitating user tasks with a high level of user satisfaction, measuring the perceived ease of use and usefulness of the adaptations and whether A-POInter addresses problems identified in existing mobile preference-based search systems such as clutter.

Objectives of the Study:

A-POInter is designed to adapt according to differing user behaviour. You will be asked to evaluate these adaptations after you have completed testing the system at the conclusion of the field study. A-POInter can adapt in various ways such as:

Information Adaptation

1. Remembering your most recently used (MRU) starting map;
2. Automatically selecting the most appropriate zoom level after running a search for POIs;
3. Automatically adjusting the level-of-detail for graphical elements on screen according to the current zoom level;
4. Suggesting search criteria according to your most frequently used (MFU) and MRU preference history;
5. Automatically filtering search results to only show the most relevant POI results;

Interaction Adaptation

1. Reordering search criteria lists by placing your preferred (MRU and MFU) criteria at the top of the list;
2. Remembering and activating your preferred zooming and panning techniques;

Visualisation Adaptation

1. Remembering your preferred map style (e.g. road map or satellite/aerial photos);
2. Automatically zooming according to your travel speed (if GPS sensor available);

Instructions / General guidelines for Participants:

The evaluation of A-POInter will last for approximately 1 week. During this period, you need to use the system as often as possible, ideally around 5 times in the week (for at least 10 minutes at a time). It is important to use A-POInter in as natural a fashion as possible, exploring maps and searching for POIs. At the end of the evaluation period, you will be asked to complete a short questionnaire measuring your satisfaction with the system.

You are expected to make use of the following main features provided by A-POInter:

1. Selecting, downloading and browsing maps
2. Manipulating maps (zooming, panning, changing map styles etc)
3. Search for POIs (e.g. restaurants, accommodation) and view and interact with results (filtering, viewing POI contact details etc)
4. (if applicable) Use the GPS to track your location / speed (whilst mobile)

Please keep in mind that A-POInter is a prototype and therefore has a very limited set of POIs upon which to search (mostly around Port Elizabeth, South Africa) and that POI information might be slightly inaccurate or out-of-date.

At the conclusion of the field study, please mark the options in the following questionnaire which most appropriately reflect your impressions about using the A-POInter system. Space at the end of the questionnaire has been provided for your written comments. If there are any ambiguities in the questionnaire please contact the investigator for a clarifying explanation.

Please email your completed questionnaire electronically to Ryan.Hill@nmmu.ac.za

Thank you for participating in this evaluation of A-POInter.

A-POInter Installation Instructions:

1. **NB:** Set your device's regional settings to English (South Africa)
    a. Go to your device start menu > Settings > Regional Settings
    b. Select "English (South Africa)" from the dropdown list.
    c. Soft-reset your device.
2. Transfer *POInterInstaller.CAB* to your device (e.g. via USB cable).
3. Browse to the *CAB* file on your device and double-click to install.
4. When prompted, install A-POInter to your **DEVICE** (i.e. not the Storage Card / External Storage)
5. Install the Microsoft .NET Compact Framework v3.5 on your device
    a. The official download is available from: http://www.microsoft.com/downloads/details.aspx?FamilyID=E3821449-3C6B-42F1-9FD9-0041345B3385&displaylang=en
    b. After downloading the *NETCFSetupv35.msi* file, run the file while your device is connected to your computer to install it to your device.
6. Transfer the 'starter' map pack to your device (ideally the storage card if present).
    a. If you have a storage card, place the provided maps folder in the following folder path on your storage card (create the folder if necessary):
    \\Storage Card\\POInter\\Cache
    (so for example, you should have a file structure like:
    \\Storage Card\\POInter\\Cache\\h0.jpeg)
    b. If you do not have a storage card, place the provided maps folder in the following folder path in the root of your mobile device (create the folder if necessary):
    \\My Device\\POInter\\Cache
    (so for example, you should have a file structure like:
    \\My Device\\POInter\\Cache\\h0.jpeg)
7. After transferring the map files to your device/storage card, proceed to launch A-POInter and follow the "first-time use" guide in the test plan provided.

a. **NB: You MUST go to "Settings" in A-POInter to configure your map cache file path location**, as well as configure your device's Internet connection type and GPS settings (if present).

If you require any further assistance with installation of A-POInter, the .NET CF v3.5, or the map pack, please contact Ryan Hill.

Contact Details:

Ryan Tebbutt Hill
Masters Student: Department of Computer Science and Information Systems
Nelson Mandela Metropolitan University
South Campus, Summerstrand
Port Elizabeth, South Africa

Email: ryan.hill@nmmu.ac.za
Web: http://www.nmmu.ac.za/ryanhill
Tel: +27-41-504-2094
Cell: +27- 82-530-2481

Instant Messaging:
Google Talk (Jabber/XMPP): tebbutt@gmail.com
Windows Live Messenger (MSN): tebbutt85@hotmail.com
ICQ: 318285628
Skype: tebbutt85
Yahoo: tebbutt1985
AIM: TeBBuTT85
MXit: 0825302481

The Department of Computer Science & Information Systems

A-POInter Questionnaire:

Nelson Mandela Metropolitan University
for tomorrow

Please complete the following questionnaire electronically by marking the appropriate box.

A. Biographical Details					
Age:					
21 - 29	30 - 39	40 - 49	50 - 59	60+	
Nationality:					
South African	International (please specify country) ………………………………				
Gender:					
Male	Female				
Computer Experience (years):					
0 - 1	1 - 2	3 - 4	5+		
PDA/Smartphone Experience (years):					
0 - 1	1 – 2	3 - 4	5+		

B. Information (Data) Adaptation					
Maps / Area of Interest					
1a) A-POInter automatically selects the most-recently-used (MRU) map	Not noticeable				Noticeable
	1	2	3	4	5
2a) A-POInter automatically selects a starting map	Not useful				Useful
	1	2	3	4	5
b)	Frustrating				Satisfying
	1	2	3	4	5
3) Selecting and browsing maps that have been downloaded	Difficult				Easy
	1	2	3	4	5
Zooming					
4a) A-POInter automatically selects a zoom-level after running a search	Not useful				Useful
	1	2	3	4	5
b)	Frustrating				Satisfying
	1	2	3	4	5
Searching and POI Search Results					
5a) A-POInter ticks my preferred criteria when searching for POIs	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Frustrating				Satisfying
	1	2	3	4	5
6a) A-POInter automatically adjusts the filter slider after a search	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Confusing				Clear
	1	2	3	4	5

d)	Frustrating				Satisfying
	1	2	3	4	5
7) Searching for POIs in A-POInter	Difficult				Easy
	1	2	3	4	5

C. Interaction (User Interface) Adaptation					
Menu and List Options					
1a) A-POInter reorders search criteria lists by placing preferred criteria at the top of the list	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Confusing				Clear
	1	2	3	4	5
d)	Frustrating				Satisfying
	1	2	3	4	5
Zooming and Panning					
2a) A-POInter remembers and automatically enables my preferred zooming technique	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Frustrating				Satisfying
	1	2	3	4	5
3) Zooming in and out of maps in A-POInter	Difficult				Easy
	1	2	3	4	5
4a) A-POInter remembers and automatically enables my preferred panning technique	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Frustrating				Satisfying
	1	2	3	4	5
5) Panning around the map in A-POInter	Difficult				Easy
	1	2	3	4	5

D. Visualisation (Presentation) Adaptation					
Map Style					
1a) A-POInter uses my preferred map-style (road or hybrid) when viewing a new map or set of search results	Not noticeable				Noticeable
	1	2	3	4	5
b)	Not useful				Useful
	1	2	3	4	5
c)	Frustrating				Satisfying
	1	2	3	4	5
2a) A-POInter automatically adjust the level-of-detail based on the zoom level (e.g. by showing the POI name at closer zoom levels)	Not noticeable				Noticeable
	1	2	3	4	5

	Not useful				Useful
b)	1	2	3	4	5
	Frustrating				Satisfying
c)	1	2	3	4	5
GPS Sensor Information (if applicable)					
	Frustrating				Satisfying
3a) A-POInter automatically adjusts the zoom level based upon the speed at which I am travelling	1	2	3	4	5
	Not useful				Useful
b)	1	2	3	4	5

E. General					
1) I would like to use the A-POInter system in the future	Strongly Disagree				Strongly Agree
	1	2	3	4	5
2) I felt in control of the system adaptations	Strongly Disagree				Strongly Agree
	1	2	3	4	5
3) I found learning to use A-POInter easy	Strongly Disagree				Strongly Agree
	1	2	3	4	5
4) I found it difficult to get A-POInter to do what I wanted it to do	Strongly Disagree				Strongly Agree
	1	2	3	4	5
5) My interaction with A-POInter was clear and understandable	Strongly Disagree				Strongly Agree
	1	2	3	4	5
6) Using A-POInter would improve my tourism experience	Strongly Disagree				Strongly Agree
	1	2	3	4	5
7) I am satisfied with the adaptations provided by A-POInter	Strongly Disagree				Strongly Agree
	1	2	3	4	5

F. Open questions

1) What was the most positive aspect of A-POInter? (which feature did you enjoy most)

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

2) What was the most negative aspect of A-POInter? (which feature did you dislike the most)………………………………………………………………………………………………

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

3) If you have any other possible adaptation ideas or suggestions for improvement, please describe these below.

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

……………………………………………………………………………………………………..

Please email your completed questionnaire electronically to Ryan.Hill@nmmu.ac.za

Thank you for your participation!

Appendix E: A-POInter Evaluation Results

Qnaire	**Noticeability**	**Usefulness**	**Satisfaction**	**Ease of Use**	**Usefulness**	**Satisfaction**	**Noticeability**	**Usefulness**	**Satisfaction**	**Noticeability**	**Usefulness**	**Ease of Use**	**Satisfaction**	**Ease of Use**
	B1A	**B2A**	**B2B**	**B3A**	**B4A**	**B4B**	**B5A**	**B5B**	**B5C**	**B6A**	**B6B**	**B6C**	**B6D**	**B7**
P01	4	5	4	3	5	4	5	5	5	4	5	4	4	3
P02	3	4	3	3	2	3	3	2	2	1	1	1	1	5
P03	3	5	4	5	5	5	3	3	3	3	4	4	4	4
P04	4	5	5	4	5	4	3	5	4	4	3	4	3	5
P05	5	5	5	5	5	5	5	4	4	5	4	4	4	3
P06	5	5	5	5	4	4	5	5	5	1	4	5	5	5
P07	4	4	4	3	4	1	4	4	3	2	2	3	3	5
P08	5	4	3	3	4	2	4	2	3	4	4	4	4	5
P09	5	4	4	4	4	4	5	5	5	4	3	4	3	5
P10	5	5	5	4	4	4	5	5	4	2	4	4	4	5
P11	5	5	5	4	4	4	5	5	4	3	4	4	4	5
P12	4	4	3	3	4	3	3	4	4	4	4	3	3	5
P13	5	4	4	3	4	4	5	5	5	3	4	4	3	5
P14	5	5	4	3	4	4	5	4	4	2	4	3	3	4
P15	5	5	5	5	5	5	5	5	5	5	4	5	4	5
Mean	4.47	4.60	4.20	3.80	4.20	3.73	4.33	4.20	4.00	3.13	3.60	3.73	3.47	4.60
Std Dev	0.74	0.51	0.77	0.86	0.77	1.10	0.90	1.08	0.93	1.30	0.99	0.96	0.92	0.74
Mode	5	5	4	3	4	4	5	5	4	4	4	4	4	5
Median	5	5	4	4	4	4	5	5	4	3	4	4	4	5

Qnaire	**Noticeability**	**Usefulness**	**Ease of Use**	**Satisfaction**	**Noticeability**	**Usefulness**	**Satisfaction**	**Ease of Use**	**Noticeability**	**Usefulness**	**Satisfaction**	**Ease of Use**
	C1A	**C1B**	**C1C**	**C1D**	**C2A**	**C2B**	**C2C**	**C3A**	**C4A**	**C4B**	**C4C**	**C5A**
P01	5	5	5	4	5	5	4	4	5	5	5	3
P02	4	3	4	2	3	3	3	4	1	3	2	4
P03	4	5	5	5	3	5	5	4	4	4	5	5
P04	3	3	4	4	5	5	5	3	5	4	4	4
P05	5	4	4	4	5	5	5	3	5	4	4	3
P06	1	5	5	5	5	4	4	4	5	5	5	3
P07	3	5	3	4	1	2	3	3	2	4	4	2
P08	4	3	4	4	4	5	4	5	3	5	4	5
P09	4	4	4	4	4	4	4	5	4	4	4	4
P10	5	5	5	5	4	4	4	5	4	4	4	5
P11	5	5	5	5	4	5	4	5	4	5	4	5
P12	4	3	3	4	5	4	4	5	5	4	4	5
P13	5	5	5	5	4	4	5	4	5	5	5	4
P14	4	5	5	4	4	4	3	4	4	4	4	2
P15	5	5	5	5	5	5	5	5	5	4	4	5
Mean	4.07	4.33	4.40	4.27	4.07	4.27	4.13	4.20	4.07	4.27	4.13	3.93
Std Dev	1.10	0.90	0.74	0.80	1.10	0.88	0.74	0.77	1.22	0.59	0.74	1.10
Mode	5	5	5	4	5	5	4	4	5	4	4	5
Median	4	5	5	4	4	4	4	4	4	4	4	4

Qnaire	**Noticeability**	**Usefulness**	**Satisfaction**	**Noticeability**	**Usefulness**	**Satisfaction**	**Satisfaction**	**Usefulness**
	D1A	**D1B**	**D1C**	**D2A**	**D2B**	**D2C**	**D3A**	**D3B**
P01	5	5	5	5	4	4		
P02	4	3	3	4	4	3	3	3
P03	5	5	5	5	5	5	5	5
P04	5	5	5	5	5	5		
P05	4	5	4	4	5	4	4	3
P06	1	5	5	5	5	5	5	5
P07	2	4	2	5	5	5	3	3
P08	5	5	4	4	5	5		
P09	4	5	5	5	5	5	4	5
P10	4	5	5	4	5	5	4	5
P11	4	5	4	4	4	5	5	5
P12	3	4	4	3	4	3	4	5
P13	5	5	5	3	5	4	5	4
P14	4	3	3	4	4	5		
P15	5	5	5	5	5	5	4	5
Mean	4.00	4.60	4.27	4.33	4.67	4.53	4.18	4.36
Std Dev	1.20	0.74	0.96	0.72	0.49	0.74	0.75	0.92
Mode	5	5	5	5	5	5	4	5
Median	4	5	5	4	5	5	4	5

Qnaire	**I would like to use the A-POInter system in the future**	**I felt in control of the system adaptations**	**I found learning to use A-POInter easy**	**I found it easy to get A-POInter to do what I wanted it to do**	**My interaction with A-POInter was clear and understandable**	**Using A-POInter would improve my tourism experience**	**I am satisfied with the adaptations provided by A-POInter**	**Age**	**Nationality**	**Gender**	**Comp Exp**	**PDA/S Exp**
	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**					
P01	4	4	5	4	4	5	4	1	2	1	4	4
P02	5	4	4	5	4	5	5	2	2	1	4	4
P03	4	3	5	5	5	3	5	1	2	1	4	2
P04	5	4	5	5	5	5	5	1	2	1	4	2
P05	3	4	4	4	3	4	4	2	2	1	3	1
P06	4	5	5	4	5	4	4	1	2	1	4	4
P07	3	2	4	2	4	4	3	2	2	1	4	4
P08	5	3	5	5	4	5	4	1	2	1	4	2
P09	5	4	4	4	4	5	5	4	1	2	4	1
P10	5	4	5	5	5	5	5	1	1	1	4	3
P11	5	5	5	5	5	5	5	4	1	1	4	3
P12	5	5	5	4	4	4	4	1	1	1	4	4
P13	4	4	5	4	5	4	5	1	1	2	4	1
P14	4	3	5	4	4	4	4	1	2	1	4	4
P15	5	5	5	5	5	5	5	4	1	1	4	4
Mean	4.40	3.93	4.73	4.33	4.40	4.47	4.47	1.80	1.60	1.13	3.93	2.87
Std Dev	0.74	0.88	0.46	0.82	0.63	0.64	0.64	1.21	0.51	0.35	0.26	1.25
Mode	5	4	5	4	4	5	5	1	2	1	4	4
Median	5	4	5	4	4	5	5	1	2	1	4	3

(this page intentionally left blank)