

Adaptive Interfaces for Mobile Map-based Visualisation

Bradley van Tonder, Janet Wesson

Department of Computer Science and Information Systems

Nelson Mandela Metropolitan University, PO Box 77000, Port Elizabeth, 6031

Tel: 041-504-2323 Fax: 041-504-2831

Email: {Bradley.vanTonder, Janet.Wesson}@nmmu.ac.za

Abstract— Mobile visualisation systems present many challenges for designers, due to limited screen space, processing power, memory and network bandwidth. These problems are particularly noticeable in mobile map-based visualisation systems, where a large information space must be visualised using a very small display area. Existing mobile visualisation systems use a variety of techniques to overcome these problems. Existing techniques, however, leave a lot to be desired. In this paper, adaptive user interfaces, which adapt to the individual characteristics of the user, are proposed as an alternative approach. A model is presented, which incorporates an adaptive user interface into a mobile map-based visualisation system. A prototype is then described, showing the successful implementation of the proposed model.

Index Terms— Mobile Visualisation, Adaptive User Interfaces, Map-based Visualisation

I. INTRODUCTION

Mobile devices are providing a platform for a new generation of information visualisation systems. Such systems provide the user with visualisations which not only take advantage of the device's mobility, but also of the range of technologies which modern mobile devices provide. These include built-in GPS, cameras and tilt-sensors which can be used to develop powerful visualisation systems which consider the user's environment and context.

Despite these obvious advantages, designers of mobile visualisation systems face a range of difficulties. These problems relate to screen size, awkward interaction mechanisms and resource limitations [6, 10]. These problems are particularly prevalent in map-based visualisations where large information spaces must be visualised while allowing the user to directly interact with the visualisation.

In this paper, an alternative approach is suggested. Adaptive User Interfaces (AUIs), which focus on adapting to the user, are suggested as a means of addressing the problems of mobile map-based visualisation (MMV). By adapting to the needs and preferences of the individual user, it is hoped that AUIs will help address the problems associated with MMV systems. We present a model for incorporating AUIs into MMV systems. A prototype system, based on this model, will also be described, demonstrating the successful implementation of the model.

This paper is organised as follows. Sections II and III discuss related work in the fields of mobile visualisation and AUIs respectively, including a discussion of existing models for AUIs. In Section IV, a model for integrating AUIs into MMV systems will be presented. The implementation of

this model in the form of a prototype will then be described in Section V, followed by conclusions in Section VI.

II. MOBILE VISUALISATION

Information Visualisation is a well-established research area which has seen a variety of useful applications. Initially, mobile visualisation was not feasible because of the limited capabilities of mobile devices. However, this is rapidly changing. Today, mobile devices have processors with clock speeds of hundreds of megahertz, high resolution displays and much greater memory capacity than in years gone by. Developers now also have access to more advanced graphical libraries, including those for 3D graphics (such as OpenGL ES) and Scalable Vector Graphics (SVG). In comparison to desktop systems, however, capabilities of mobile devices still lag far behind.

A. Problems and Shortcomings

Due to their compact size, mobile devices lack the performance capabilities of desktop computers. In order to accommodate this, mobile visualisation techniques need to be adapted. Several shortcomings of mobile devices have been identified and need to be considered when designing mobile visualisation systems. These include relatively slow hardware, small displays, limited battery life and input peripherals that are awkward to use [3, 6].

With ongoing improvements in technology and manufacturing processes, it is likely that mobile device capabilities will continue to improve in the years to come. Nevertheless, many of these problems, such as limited screen space, are likely to remain.

B. Existing Techniques

Many of the existing techniques used by mobile visualisation systems have their roots in desktop visualisation systems. These techniques have been adapted for mobile devices in efforts to address the above problems that designers of such systems face. Despite this, many of these techniques introduce problems of their own.

- Hierarchical organisation [17] involves splitting information into a hierarchy of screens, each small enough to fit onto the screen. This can result in the user having to navigate a complex hierarchy of views in order to find information of interest.

- Zooming and panning [6] allows the user to navigate views which are larger than the screen by panning them, as though viewing them through a portal, and then zooming in and out as necessary. This can result in a lot of interaction being necessary from the user, who may also lose track of how the current view relates to the bigger picture.

- Overview+Detail is a technique which dedicates a portion of the display to showing how the current visualisation fits into the broader context [6]. On mobile

phones, however, this overview can be illegible due to its small size.

- Focus+Context is a technique which has been adapted for use in mobile devices. Most applications of Focus+Context in desktop systems use a Fisheye lens approach, where the currently visualised area is in focus and surrounding areas are distorted on the periphery [9]. In contrast to Overview+Detail, both the overview and detail views are integrated into a single view, showing how the detail fits into the broader context. Focus+Context techniques are often resource intensive and are not suited to all data types, particularly map visualisations, where they can distort the user's perception of distance.

- Several other techniques have also been developed which visualise the location of off-screen objects. Halo surrounds off-screen objects with circles just large enough to encroach onto the currently focused area, allowing the user to determine the distance and direction to off-screen objects [2]. CityLights uses points, lines or arcs on the border of the display to denote the direction and distance to off-screen objects [20].

This overview of mobile visualisation techniques shows that no perfect solution for mobile visualisation currently exists. New techniques attempt to solve the problems of older techniques, only to introduce problems of their own. Adaptive visualisation techniques represent a possible means of addressing the problems of existing techniques.

III. ADAPTIVE USER INTERFACES

An Adaptive User Interface has been defined as [12]:

“An adaptive user interface is a software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user.”

The basic premise behind AUIs is that users are different and therefore have different needs from an interactive system. The system should adapt to the user, rather than forcing the user to adapt to the system. AUIs rely on a variety of models to improve their interaction with the user, of which the User Model is the most important. Each user's characteristics and/or past behaviour are modelled and stored in the User Model in an attempt to adapt to his/her needs and desires. AUIs have been identified as consisting of the following three high-level components [15]:

- *Afferential component*: Observes and records user behaviour and system reactions;
- *Inferential component*: Analyses the gathered data to draw conclusions and decide how the system should adapt to the User Model; and
- *Efferential component*: Adaptations lead to modifications of how the system behaves.

Adaptation has previously been recognised as a useful means of overcoming the limitations of mobile devices [1]. Hence it is reasonable to assume that AUIs could provide the means to overcome some of the problems and shortcomings of mobile visualisation.

A. Adaptive Visualisation Systems

AUIs exist in many different forms, ranging from those that make sweeping interface changes to recommender systems that have proven successful in e-commerce applications. The class of mobile visualisation systems which has seen the most attention in terms of the application

of AUIs, are those which present map-based visualisations to the user. Such systems are among the most complex of mobile visualisation systems, visualising large volumes of data. MMV systems are thus the focus of the model to be presented in this paper.

Several MMV systems have been developed which provide varying levels of adaptation in terms of the maps themselves and the data visualised. *Gulliver's Genie* is a personalised mobile tourist guide that attempts to overcome bandwidth limitations by pre-caching content that the user may wish to view, given his or her context and preferences. Issues of switching output modalities and map rotation are also addressed and personalised for the user [14]. Other authors have detailed more holistic approaches to adapting maps and map-based visualisations, attempting to minimise the disadvantages of mobile devices by adapting to the individual user [10]. Several aspects of MMV systems have been targeted for adaptation, including adapting the level of detail with which the geometry of map objects are rendered and adapting MMV systems to different devices with differing capabilities.

IV. EXISTING MODELS

Many existing AUI systems have followed an ad-hoc approach to development, with different developers incorporating very different models in the design of their individual systems. Nevertheless, a variety of general models, frameworks and architectures have been developed and two of these form the basis of the model proposed in Section V.

A. The Mobile Cartographic Framework

Reichenbacher [16] proposed a framework for AUIs for mobile cartographic systems. This framework is particularly relevant for this research because it combines mobile visualisation and adaptation. The framework can be seen in Figure 1.

Reichenbacher identifies four different areas in which adaptation is possible:

- Information content;
- User interface;
- Visualisation; and
- Technology (adapting to different device capabilities).

Adaptation takes place in response to changes in the user's context. If the change in user context exceeds a certain threshold, then adaptation may be necessary in one of the four adaptation areas.

The Mobile Cartographic Framework is limited to the variable of context due to the fact that it is designed for mobile cartographic applications only. Neither a User Model nor a Task Model (which are regarded as essential components of AUIs [11]), is present in this framework. As a result, any system based on this framework would not be able to adapt to previous user behaviour or user characteristics. In order to see how these models can be integrated into an AUI, the next section will look at an existing AUI framework which incorporates a variety of different models.

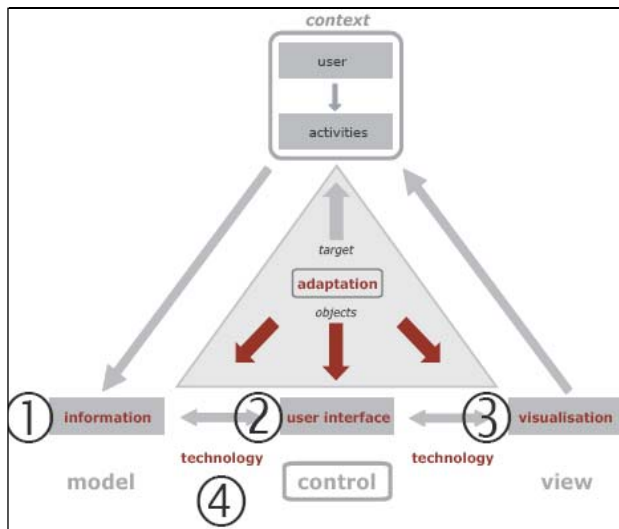


Fig 1. The Mobile Cartographic Framework [16]

B. The AeS Framework

The framework for Adaptive eHealth Systems (AeS) is specifically designed for adaptive eHealth systems [19]. The framework is one of the best examples of an AUI framework, with a Knowledge Base which includes a wide variety of models, facilitating a broad range of adaptations. The User Model and Task Model, which have been identified as essential AUI components [11] are present in this framework, unlike the Mobile Cartographic Framework. An Environment Model is also included, and appears to be a hybrid of the System and Domain Models typically found in AUIs, modelling the current system status and knowledge relevant to the particular application domain. These models are used in the framework to make adaptations to the content, presentation and navigation of the system, and provide the means to facilitate a wide variety of adaptations. A new model is needed which enhances the Mobile Cartographic Framework by incorporating the additional AUI models in the AeS framework.

V. PROPOSED MODEL

In this section a new model is proposed, which incorporates the Knowledge Base of the AeS framework into a new model for adaptive MMV, called the Proteus Model. The architecture of the Proteus Model is shown in Figure 2. Similar to the Mobile Cartographic Framework, the Proteus Model focuses on adaptations in three main areas:

- *Information* – adapting *information* selection and organisation, in order to provide the user with the most relevant data, according to his/her User Model and minimise user interaction;
- *Visualisation* – adapting the *presentation* of the visualised information according to the user's preferences; and
- *Interface* – adapting the *user interface* through which the user is able to manipulate the visualisation.

A. Components

1) Data Model

The Data Model contains the data which is being visualised by the system (Figure 2). In MMV systems, this is split into map data and additional data which is overlaid or visualised in the context of the underlying map. The map data may

consist of raw data (e.g. points, lines and features) from which the maps are rendered, or the maps could be downloaded as image tiles. The additional data depends on the domain of the particular system. For example, it could consist of points of interest in a system to help tourists find landmarks or hotels, or the additional data could show levels of traffic congestion super-imposed on a map of a city.

2) System Model

The System Model maintains the current state of the system (Figure 2). In order to manage adaptation of different parameters, it is necessary for the system to be aware of the current values of these parameters. As both the user and system are able to make changes to these parameters, a central repository is needed which stores the current system state. Interaction data must be processed in order to determine if changes to the System Model are necessary.

3) Task Model

Adaptations such as task assistance, interface adaptation and dynamic presentation can be improved if the system is able to recognise what the user is trying to do. To achieve this, a Task Model is necessary (Figure 2). Such a model would contain typical user tasks, as well as the various steps involved in achieving each task. User goals could then be identified by comparing observed user actions with those stored in the Task Model.

4) User Model

The User Model is the most important model and contains all the knowledge that the system has acquired regarding the user and his/her preferences for different tasks.

5) Context Model

Time and location are two important components of MMV systems, particularly those that provide visualisations in real time. Knowledge of these two elements is potentially valuable in informing the adaptation process and is stored in the Context Model (Figure 2).

6) User Monitoring and Modelling Component

The User Monitoring and Modelling Component (UMMC) is responsible for accepting user interaction data as input and converting this to knowledge regarding the user's preferences and behaviour (Figure 2). The Proteus Model does not prescribe a particular user modelling technique, as different techniques may be suited to different situations. The user modelling technique is responsible for using user interaction data to refine an initial default User Model.

7) Visualisation Manager

The Visualisation Manager is responsible for managing changes to the visual representation of the information (Figure 2). This is done in consultation with the User and System Models. A number of different visualisation parameters of MMV systems can be adapted, including detail level and zoom level.

8) Information Manager

The Information Manager filters and organises the information to be visualised and presented to the user (Figure 2). The User Model is consulted to determine the user's preferences.

9) Interface Manager

The Interface Manager is responsible for making changes to user interface controls such as menu options (Figure 2). The User Model is consulted to order lists of options so that the items likely to be of interest to the user are at the top, and those of little or no interest at the bottom. Shortcuts available to the user could also be adapted.

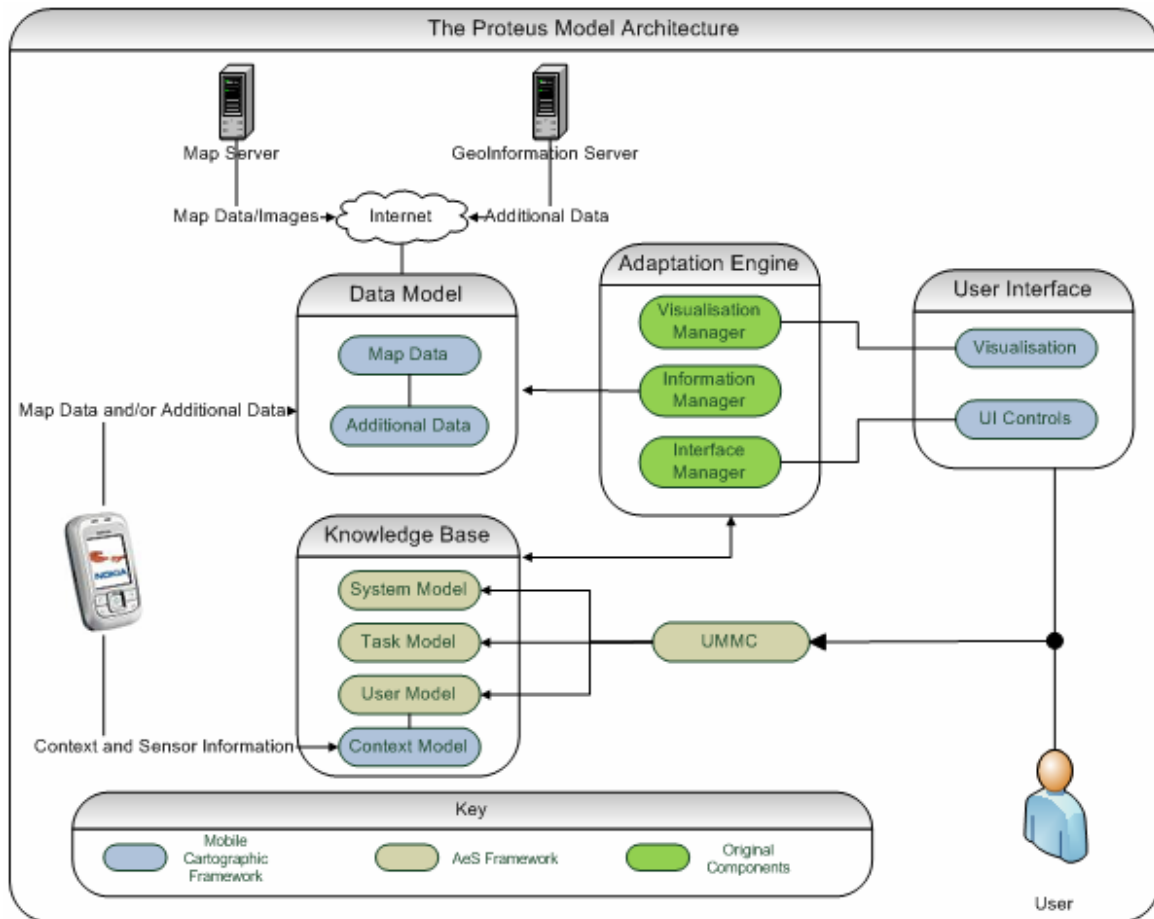


Fig 2. Proteus Model Architecture

B. Adaptation Timing

Adaptation in MMV systems needs to be designed so that it does not slow down the use of the system by monopolising the limited CPU power and memory. In order to do this, the Proteus Model advocates an approach where user modelling and updates to the User Model are performed only when the system is shutting down. When the system is started up, the User Model is loaded into memory. Adaptation is only performed when a new task is started. For example, if the user selects an option that will result in the generation of a new visualisation, the Visualisation Manager is invoked to ensure that the visualisation is rendered in accordance with the user's preferences. The Visualisation Manager then consults the System and Task Models in order to determine the current user task. The user's preferences for that specific task are then retrieved and used to ensure that the visualisation is rendered in accordance with the user's preferences. The Interface and Information Managers work in a similar fashion. User interaction data is recorded during system use and only written to a data file when the system shuts down or when a certain threshold is reached.

VI. IMPLEMENTATION

In this section, the implementation of a prototype application based on the Proteus Model is presented.

A. Application

A prototype mobile application, named *MediaMaps*, was implemented in Java Micro Edition. *MediaMaps* is a mobile application which allows the user to capture and geo-tag (with latitude and longitude information) photographs, videos and sound recordings. *MediaMaps* is also responsible for grouping these multimedia items into "collections" using information adaptation. These collections are intended to cluster together multimedia items corresponding to the same events or closely related events (Figures 3 and 4). This is to prevent users from having to manually organise their media collections and allows related items to be visualised together.

MediaMaps then allows the user to visualise this information in a map-based context using information, visualisation and interface adaptation. This prototype implements the different components of the Proteus Model, adapting in the three areas identified in the model, namely information, visualisation and interface.

This application was chosen to be implemented because it possesses the characteristics of a typical MMV system. Visualisation of location-tagged pictures has become a popular trend on the Web in recent times, with photo-sharing websites such as *Flickr* and many others containing thousands of location-tagged images uploaded by users [18]. Research has shown that users are able to recall the location where a photograph was taken more accurately than other variables [13], making map-based browsing of photo and other media collections a useful means of browsing and searching media stored on the mobile device itself.

B. Adaptation Techniques

1) Information Adaptation

Information adaptation takes place both at the time of capture and during visualisation in *MediaMaps*. Information adaptation is achieved during the capture of new media through the implementation of the Real-time-Event Detection (RED) algorithm [5], which is discussed in Section C below.

Information adaptation also takes place during visualisation. This includes adapting the time period, the initial location and the data types being visualised. These adaptations function similarly to the visualisation adaptations described in the following section.

Information adaptation is facilitated by the Information Manager, which is activated when new media is captured and when a new visualisation is about to be rendered in *MediaMaps*. The Task Model is used to identify when the Information Manager needs to be invoked and the System Model stores the relevant system state. The User Model is consulted when performing information adaptation in *MediaMaps* in order to determine user preferences (Figure 2).

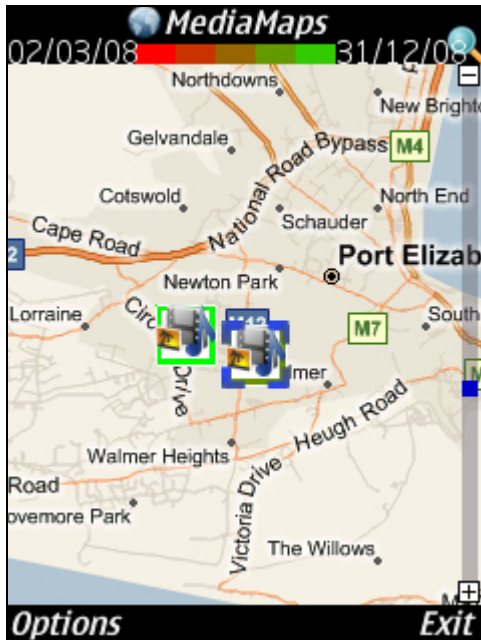


Fig 3. *MediaMaps* showing two different multimedia collections being visualised.

2) Visualisation Adaptation

Visualisation and information adaptation in *MediaMaps* is done using a set of naïve Bayesian classifiers (Section C). Visualisation Adaptation is facilitated by the Visualisation Manager, which is invoked whenever a new visualisation is to be rendered in *MediaMaps*. The System, Task and User Models are consulted as part of the adaptation process (Figure 2).

3) Interface Adaptation

Interface Adaptation in *MediaMaps* is limited to ordering of menus and lists of options. This is done using the base adaptive algorithm for ordering of menu options outlined in [6], as discussed in Section C below. These preferences are persistently stored in the User Model. Interface Adaptation is facilitated by the Information Manager component in conjunction with the models in the Knowledge Base (Figure 2).

C. Adaptation Algorithms

1) The RED Algorithm

The RED algorithm was developed to automatically cluster photos into related events [5]. (In *MediaMaps*, this was extended to all multimedia types.) This is done by assuming that the user's photo-taking behaviour approximately follows a Poisson distribution. A Poisson process is a counting process expressed as a function of time, with four basic properties:

- Occurrences of events are independent;
- Simultaneous events are not possible;
- It is stationary; and
- Where time = 0, the count is also 0.

Confidence thresholds are used to determine whether a new media item (photo, video or sound recording) belongs to the current collection or a new collection. This initial test is based on the time when the media item was recorded. If the item falls into an intermediate zone between new and old collections, the item's location is retrieved to help determine whether the current item belongs in the current collection, or if a new collection should be created. This process is done incrementally and in real-time, consulting the Context Model and Task Model to aid in the classification process.

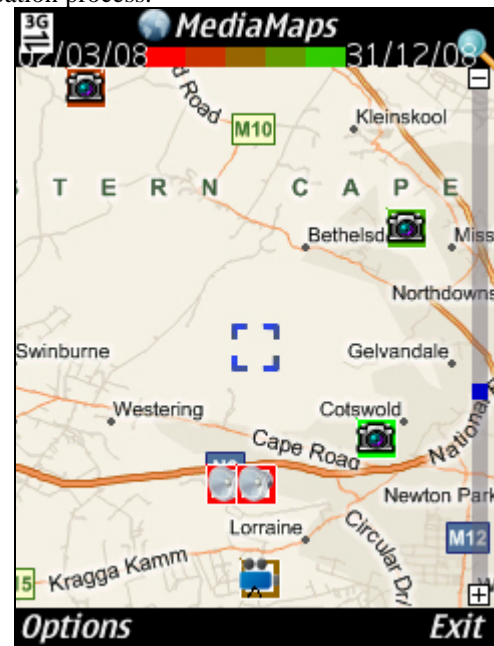


Fig 4. *MediaMaps* showing individual media items from the highlighted collection in Fig. 3

2) Naïve Bayesian Classifiers

Naïve Bayesian classifiers represent a supervised learning technique that is very simple, yet has proven highly effective [7]. They rely on the assumption that the different attributes used to classify an input are independent of one another. Although this assumption is usually violated, Bayesian classifiers often outperform more complex supervised learning techniques. They are particularly appealing for the user modelling process in a MMV system such as *MediaMaps* because of their simplicity, and hence the efficiency with which they can calculate the user's preferences. They have also been successfully used in the past in order to adapt interfaces in a mobile context [4]. Bayesian classifiers calculate the probability that a tuple belongs to a certain class given the values of certain input

attributes. Previous user behaviour, recorded by the UMMC, can be used to train the classifiers. If, for example, we were calculating the user's preferred zoom level, (based on the current collection and mode), the formula would look as follows (probabilities would have to be calculated for all possible zoom levels and the maximum value determined):

$$P(\text{Zoom} = 11 | \text{collection}=1, \text{mode}=\text{animation}) = \frac{P(\text{collection}=1, \text{mode}=\text{animation} | \text{Zoom}=11) \cdot P(\text{Zoom}=11)}{P(\text{collection}=1, \text{mode}=\text{animation})}$$

Similar classifiers are used for all the different display parameters, as well as for determining the data types to display.

3) Base Adaptive Algorithm

This algorithm dedicates the first few positions in the menu or list to the most recently selected and most frequently selected options. This ensures that the user's most likely choice remains at the top of the menu.

VII. CONCLUSION

This paper has proposed a model for designing adaptive user interfaces for mobile visualisation. Existing visualisation techniques were shown to be inadequate. The use of AUIs for MMV represents a new approach which could potentially address the shortcomings of existing techniques. The Proteus Model was proposed to support a wide range of adaptations. This model can be used to facilitate the development of MMV systems which use adaptation in order to optimise user interaction. A prototype mobile application, called *MediaMaps*, was developed as a proof of concept of the Proteus Model. This application uses information, visualisation and interface adaptation to allow users to capture and browse collections of media items in a map-based context. Future work will include a field study to evaluate the adaptive features of *MediaMaps*.

ACKNOWLEDGEMENT

Acknowledgements are due to the Telkom Centre of Excellence Programme and the NMMU Department of Computer Science and Information Systems for making this research possible.

REFERENCES

- [1] Al-bar, A. and Wakeman, I. A Survey of Adaptive Applications in Mobile Computing. In *Proceedings of International Conference on Distributed Computing Systems (ICDCSW '01)* (2001). 246.
- [2] Baudisch, P. Halo: A Virtual Periphery for Small Screen Devices. In *Proceedings of Advanced Visual Interfaces (AVI '04)* (2004). 80-84.
- [3] Björk, S., Redström, J., Ljungstrand, P. and Holmquist, L. Powerview: Using information links and information views to navigate and visualize information on small displays. In *Proceedings of the 2nd international Symposium on Handheld and Ubiquitous Computing* (Bristol, UK, September 25 - 27, 2000). Springer-Verlag, London, 46-62.
- [4] Bridle, R. and McCreath, E. Inducing Shortcuts on a Mobile Phone Interface. In *Proceedings of IUI '06* (2006). 327-329.
- [5] Chen, W. and Chen, M. Event Detection in Real Time on a Personal Camera Phone Image Collection, *Multimedia Systems*, Springer-Verlag. (2007) 12, 479- 492.
- [6] Chittaro, L. Visualizing Information on Mobile Devices. *IEEE Computer*. (2006) 39 (3), 40-45.
- [7] Elkan, C. Naive Bayesian Learning. (Technical Report CS97-557). University of California, San Diego (1997).
- [8] Findlater, L. and McGrenene, J. Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces. In *Proceedings of CHI 2008* (2008), 1247-1256.
- [9] Furnas, G. Generalized Fisheye Views. In *Proceedings of SIGHCI Conference on Human Factors in Computing Systems*, Boston, Massachusetts, USA, (1981).
- [10] Hampe, M. and Paelke, V. Adaptive Maps for Mobile Applications. In *Proceedings of Mobile Maps Workshop at MobileHCI '05*. (2005).
- [11] Krogsæter, M. and Thomas, C. Adaptivity: System-Initiated Individualization. *Adaptive User Support*. R. Opperman. Hillsdale, New Jersey, Lawrence Erlbaum Associates Publishers. (1994), 67-96.
- [12] Langley, P. User Modeling in Adaptive Interfaces. In *Proceedings of Seventh International Conference on User Modeling* (1999). 357-370.
- [13] Naaman, M., Harada, S., Wang, Q., Garcia-Molina, H. and Paepcke, A. Context Data in Geo-Referenced Digital Photo Collections. In *Proceedings of 12th ACM International Conference on Multimedia* (2004). ACM, New York 196-203.
- [14] O'Grady, M.J. and O'Hare, G.M.P. Enabling Customized and Personalized Interfaces in Mobile Computing. In *Proceedings of the 9th international Conference on Intelligent User Interfaces (IUI'04)* (2004). 292-294.
- [15] Opperman, R. Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. (1994).
- [16] Reichenbacher, T. Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices Technischen Universität, München (2004).
- [17] Sanna, A. and Fornaro, A. IMoViS: A System for Mobile Visualization of Intrusion Detection Data. *Information & Security*. (2003) 12 (2), 235-249.
- [18] Torniai, C., Battle, S. and Cayzer, S. Sharing, Discovering and Browsing Geotagged Pictures on the Web. (Technical Report HPL-2007-73). HP Laboratories, Bristol, (2007).
- [19] Vasilyeva, E., Pechenizkiy, M. and Puuronen, S. Towards the Framework of Adaptive User Interfaces for eHealth. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems* (2005). 139-144.
- [20] Zellweger, P., Mackinlay, J., Good, L., Stefik, M. and Baudisch, P. City Lights: Contextual Views in Minimal Space. In *Proceedings of CHI '03 Extended Abstracts on Human Factors in Computing Systems* (2003). 838-839.

Bradley van Tonder received his B.Sc (Honours) degree in 2006 from the Nelson Mandela Metropolitan University. He is currently studying towards an M.Sc in Computer Science and Information Systems at the same institution. His research interests include mobile computing, information visualisation and adaptive user interfaces.