Automatic Generation of User Centric Interfaces for J2ME Enabled Handheld Devices

Ipadeola A.O, Olugbara O.O, Adigun M.O, Xulu S.S.

Abstract—Mobile computing is enabling an unrestricted accessibility to applications and services. The technology allows information accessibility through the use of mobile or personal communication devices (PCD). However, a typical mobile computing environment is still plagued with some challenges. Handheld devices are heterogeneous in nature, limited in computing power and characterized with small display screen sizes and small memory. The diversity in user needs and preferences coupled with variability of execution environment are additional challenges that must be resolved in order to achieve effective information communication. This work describes a Custom Made (CoMADE) toolkit for dynamically generating user centric interfaces for service consumers and providers in a mobile computing environment. CoMADE generates on-demand user interfaces for individual users based on user preferences and service request information. With CoMADE, direct user participation in user interface generation with a high degree of dynamism and flexibility were achieved. Additionally, CoMADE supports ease of application extensibility.

Index Terms— Mobile Computing**,** User Interface, Multi-Device interface, handheld devices.

I. INTRODUCTION

Advancement in wireless technology has paved way for mobile computing, which can be described as technology 'untethered', allowing transmission of data through a computer, without a connection to a fixed physical link [1]. Mobile or personal communication devices (PCD) are useful instruments for mobile information communication. They enable mobile access to applications and services. Mobile computing has immensely affected the way information is represented, transmitted or received. It has brought about significant impacts to individuals, organizations and institutions [2]. The technology allows a tourist to access his/her location based mobile services and a service provider to monitor the usage of his/her services on handheld devices [3].

In general, it brings the world closely together. However, effective information communication is hampered by some challenges. PCD are heterogeneous, they have limited computational resources, limited storage, small display sizes and are often faced with changing execution environments [4]. A second dimension of the challenges of a mobile computing environment is the heterogeneity of users. Users have diversified needs, which can be defined in terms of their demographics, information seeking strategy and purchasing intent factors [5]. Furthermore, some users require special consideration, for instance, users with disabilities [6]. The heterogeneity of devices and the diversity in user needs and preferences must therefore be addressed during user interface generation in order to achieve effective information communication. A user interface has been defined as a computer-mediated means of facilitating communication between human being(s) and an artifact [7].

The conventional approach for addressing the unique characteristics of every computing device is by designing a separate user interface for every device [8]. This method is considered inefficient due to its gross disadvantages. It is time consuming and error-prone. It produces inconsistent interfaces since the design of user interfaces for diverse devices is handled differently. In addition, this approach is not cost-effective [9]. An automatic translation method was proposed for automatically generating user interfaces for diverse device profiles. It involves designing of a core logic of an application in some flavor of eXtensible Markup Language (XML).The core logic is adapted for diverse devices [10].

In order to achieve universal accessibility and usability of information, user interfaces must be designed to suit the device or platform, the user and the execution environment. A platform is modeled in terms of resources, which in turn determines the way information is computed, transmitted, rendered and manipulated by users. An environment is a set of objects, persons and stochastic events that are peripheral to the current activity, but may have an impact on the system and/or user's behavior, either now or in the future [11]. A user can be any person who interacts directly with a computer system.

II. RELATED WORK

The universal accessibility and usability of information has necessitated for efficient mechanisms for user interface design. Basically, the model based approach has been widely adopted for user interface design for handheld devices due to its advantages.

The model based approach shields user interface designer from the implementations details [12]. The work reported in [13] describes a model based tool called DAMASK for multi-device interface design. DAMASK uses a hybrid of design pattern and model based approach to realize multi-device interfaces. The TERESA tool achieved multi-device interfaces through the use of multiple logical descriptions and a transformation algorithm. The work reported in [14], [15], [16] and [17] developed different software tools for the generation of multi-device user interfaces based on the model based approach.

However, it is very significant to address the diversity in user needs and preferences during user interface generation. In the light of this, the work in [18] proposed the consideration of the user perceptual preference characteristics during multi-device user interface design. An explicit user model was adopted in [19] for achieving a user-adapted interfaces. In [20], user characteristics were broadly classified into application dependent and independent characteristics. Application independent characteristics include preferences, capabilities and psycho-motor. Application dependent characteristics include user goals and the knowledge of the system and applications.

Most of the existing work however focused on browser based environments, such as Wireless Application Protocol (WAP) enabled phones. Information access through a browser based environment is faced with some challenges. User interfaces designed based on browser based environment are not multi-device capable. In order words, they do not support diverse devices. Incessant network disconnection is another challenge facing a browser based environment [21][22]. The J2ME platform however, offers significant and more appealing benefits to users of mobile applications and services. J2ME platform is cross platform/device compatible. It also provides proficient security system, disconnected access and synchronization provisioning, dynamic delivery of applications and services, enhanced user experience, scalability and performance [22]. This work presents a toolkit named CoMADE for dynamic generation of user centric interfaces for J2ME enabled handheld devices.

The rest of this paper is succinctly described as follows. Section III gives a comprehensive description of user centric interface generation using the CoMADE toolkit. Section IV considers the implementation of CoMADE. Section V describes the application development and evaluation details of CoMADE. Section VI gives a brief conclusion.

III. User Centric Interface Generation Using the CoMADE Toolkit

The CoMADE environment was developed for an on-demand generation of user centric interfaces for service consumers and providers in a mobile computing environment. CoMADE is based on a polymorphic logical description [23] and it is made up of three tiers namely the design, load and runtime (Fig. 2). During the design time, a polymorphic task, abstract and concrete interface generation is realized with CoMADE. The final user centric interface is realized during the load time while user actions, such as change in preferences or user requirements characterizes the run time.

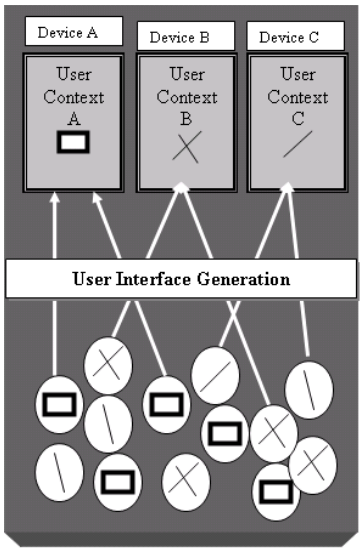

Fig. 1 User Interface Generation by Dynamic Selection of Artifacts

CoMADE considers every element of an interface as an artifact and the selection of an artifact during interface generation is based on some real time or pre-recorded user information. The user information is a set of user preferences, requirements, needs or user attributes. Fig. 1

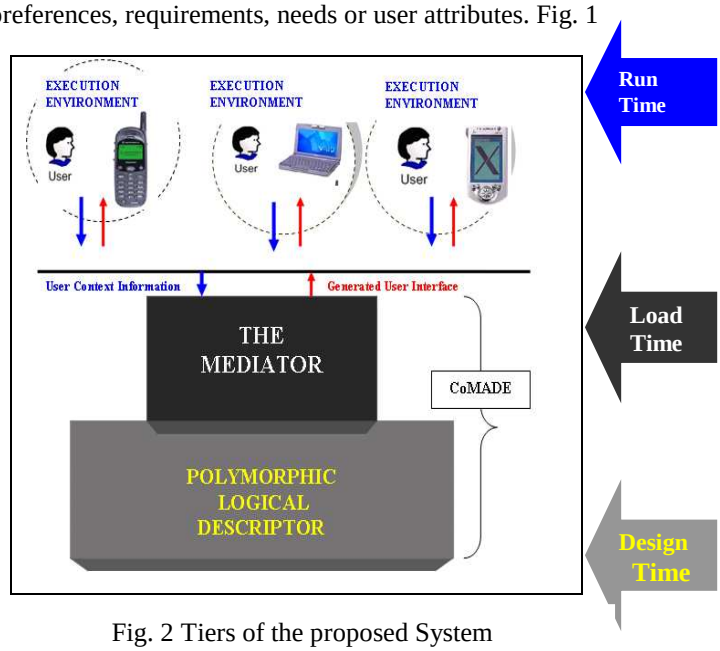

Fig. 2 Tiers of the proposed System

shows the dynamic selection process of the CoMADE toolkit. The initial step towards dynamic generation of an interface is the creation of polymorphic task model during the design time. Polymorphic task model is based on hierarchical task decomposition [24]. A user interface designer uses CoMADE for the specification of user tasks to be represented on an interface. A task is an activity to be performed by a user in order to achieve a goal [25]. The designer creates alternative representation of tasks called task variants. For instance, a task, such as "Select Your Favorite Meal" can be alternatively represented in different languages, such as Africaans and isiZulu. The polymorphic representation of task was used in order to cater for the diversities in user preferences and requirements during the design time. Every variant is associated with an attribute-value description. For instance, a variant v_1 of task T_1 can be suitable for a user and variant v_2 of the same task T_1 can be selected for another user during user interface generation. Task variants information is persisted in xml based language called the Task variant Description Language (TVDL).

The polymorphic abstract interface is automatically generated by CoMADE. It is a device-independent description of an interface. The task variants are represented in terms of their abstract interaction object (AIO). For instance, a task such as "Select Your Favorite Meal" is associated with a "SelectObject" AIO. Additionally, the task variants are structured in a presentation task set (PTS). PTS is a window or a set of task variants perceivable by a user on

his/her device at a given time. In essence, the polymorphic abstract description contains all possible user interface presentations with their associated connections. The presentation sets are constructed with associated connections from the TVDL. Every presentation set contains user tasks and their associated variants. The abstract description realized at this phase is represented in a Presentation Set Description Language (PSDL).

The polymorphic concrete interface is a device dependent user interface description. It is the concrete representation of of AIOs in the PSDL. In other words, every variant representation stored in PSDL is converted to a concrete interaction object (CIO), represented as $_{V}$CIO. For example, a 'SelectionObject' is associated with an implementation method for a ChoiceGroup interactor while an implementation method for a StringItem is generated for TextDisplayObject. J2ME was chosen as the target platform because of its unique advantages. CoMADE therefore, generates a supporting implementation method for every $_{V}$CIO object. The method generated for every $_{V}$CIO, coupled with its description in the PSDL is persisted in a database object. Specifically, an object database, db4o [26] was adopted for persisting the concrete description associated with each variant. For n set of vCIOs objects, there is an equal number of m sets of concrete description referred to as implementation methods or application dependent method (ADM). In this phase, CoMADE also generates class templates for every user interface. Class templates include, class definitions, application independent method (AIM) definitions. These aforementioned definitions are specified at design time and persisted as objects into db4o database. The designer can modify and update all generated interface component defined at this phase. Additionally, different designers or service developers can work independently on different implementation methods of the CIO variants generated by CoMADE.
The proposed toolkit also supports modularity, whereby different modules created by software experts are associated with variants of CIO. During this phase, CIOs are further enabled with functionalities. Thus, the generated CIO methods and properties can be modified. In CoMADE, a work-board is provided to enable designers or mobile web service developers import and edit additional files

Consequently, every modification is updated. The output of the concrete interface is an object containing variants of CIO implementation methods (vCIO) and class template definitions which is persisted in the object database (Fig. 3).

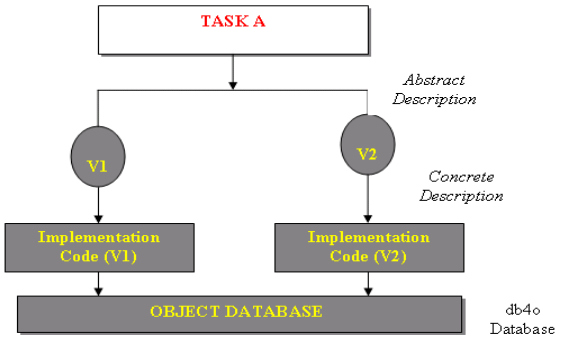

Fig. 3 Polymorphic Concrete Model Persisted Into Db4o

The mediator handles the final user interface generation. It generates user interface based on user preference information. A request can be made either to a mobile device or through a personal computer. When a request is received by the mediator, the mediator queries the information repository for the selection of a matching object database (OD). When the OD is found, the preference information is used for dynamic construction of a variant selection query. The query is performed against the OD for the selection of suitable variants. A variant is termed suitable if its attribute-value description corresponds to the user request. In the OD, every PTS is queried for the selection of suitable variants. The selected variant is accompanied with its implementation description. The mediator therefore achieves a set of User centered PTS (UPTS) from the set of PTS persisted in the OD. A collection of UPTS make up a whole interface. Furthermore, the class template for the requested application is retrieved at this phase. A set of application independent methods and other enabling modules provided by the designer, are also retrieved. Additionally, a declaration definition and order of presentation is generated during this phase. Media content adaptation is also carried out at this phase depending on user's preferences. The set of UPST, which makes up a whole user centric interface is compiled, pre-verified and packaged for a requesting user.

IV. Implementation of CoMADE

CoMADE was developed in Java and it mimics existing Integrated Development Environment. Much work in CoMADE are automated, the creation of the abstract interface, concrete interface and the persistence of concrete interface artifacts into the repository are all handled automatically. CoMADE has a work board where designer can modify or create additional interface artifacts. The toolkit persists interface description in flavours of xml similar to other model based tools. Other components like the mediator and emulator programs are integrated with CoMADE for handling load and runtime activities. Specifically, Sun emulator programs were integrated with CoMADE. During the design time, the designer can specify an example preference and then activate the mediator. The mediator generates a user centric interface and loads the user interface on an emulator program.

V.APPLICATION DEVELOPMENT AND EVALUATION

A. Application Development

We have developed an application that enables service providers to access their service usage information. The preferences of the providers were used in the generation of user centric interfaces. The mode of report generation and the language for interface presentation differ from one provider to another, basically English, isiZulu and Africaans languages were considered for user interface presentation. Service usage reports were in charts and tables. Fig 5 shows a snapshot of an interface generated in English Language and report generated in a tabular form. Fig 6 shows a snapshot of a user interface generated in isiZulu language and a service usage report generated using Charts.

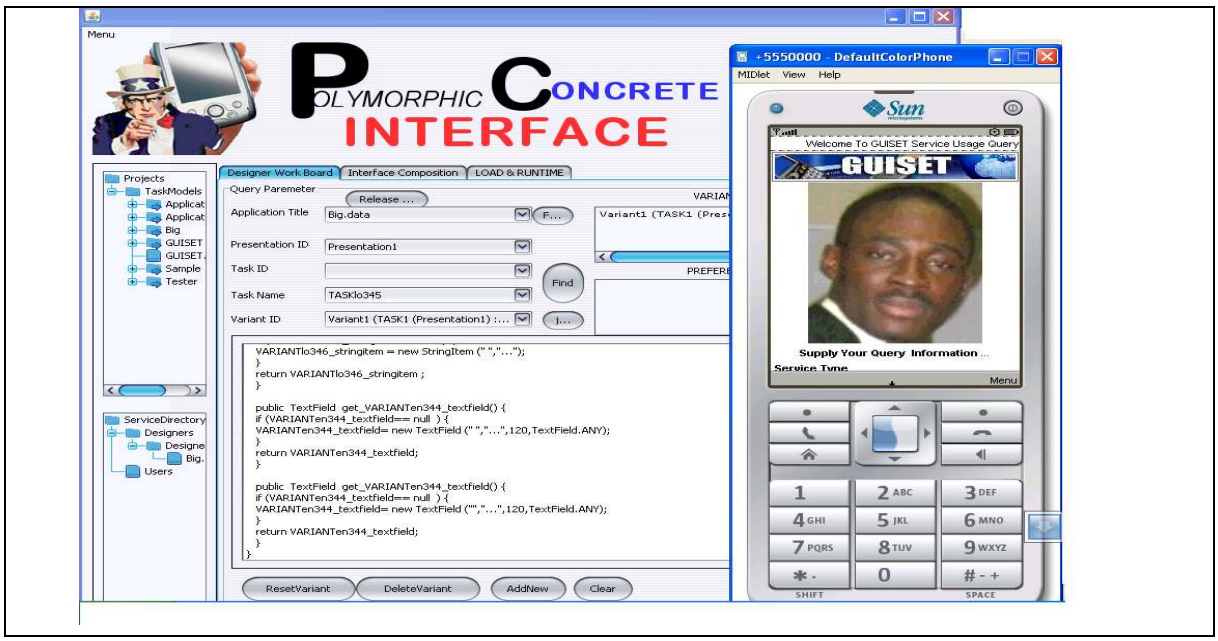

Fig. 4. Snapshot of CoMADE during Interface Generation

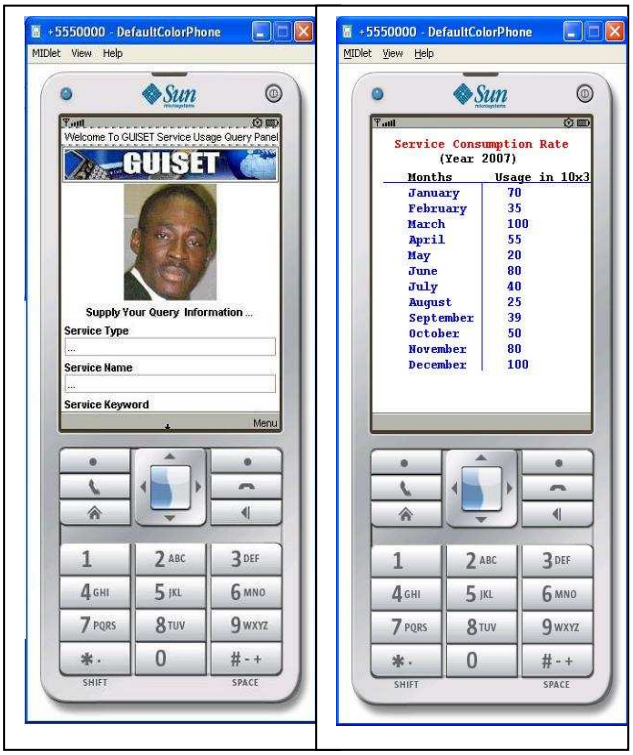

Fig. 5. User's Choice - Lang: English, Image: Colored, Report: Tables

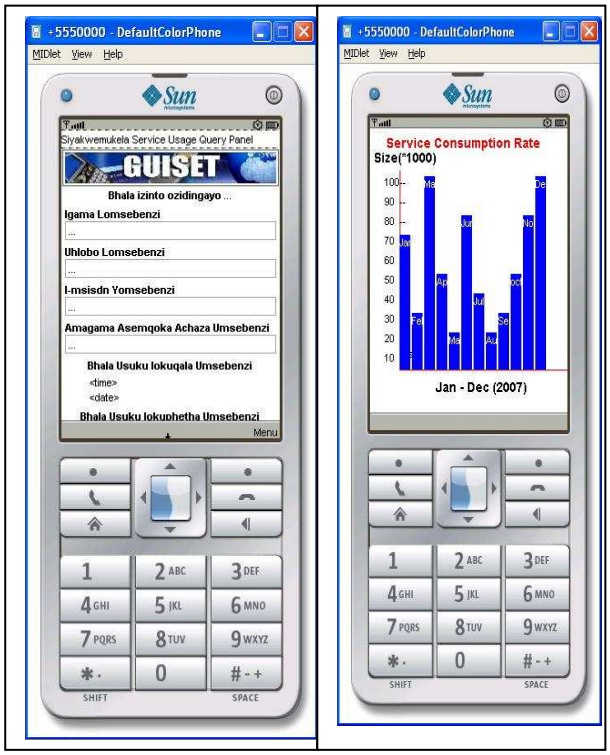

Fig. 6. User's Choice - Lang: isiZulu, Image: Non, Report: Charts

B. Evaluation

We have evaluated CoMADE toolkit based on its ability to create a polymorphic task model based on an explicit user model during the design time. The ability of CoMADE to transform the specified variants into their CIO representations was evaluated. Additionally, the ability of CoMADE to generate interfaces based on user preferences was evaluated. Three subject groups of user interface designers participated in the evaluation, namely, the expert, the intermediate and the novice designers. The usability evaluation was carried in accordance to the procedures proposed by Nielsen [27]. Two questionnaires were administered. The first questionnaire was administered prior the evaluation. It was used for categorizing the subjects into their respective groups. A video presentation, which demonstrated with an example how CoMADE can be used for user interface generation was presented. Subjects were allowed to develop a sample application using CoMADE. A second questionnaire was administered for usability testing. Additionally, it was used also used to evaluate the proposed CoMADE. Fig. 7 shows that 14.29%, 30.77%, 75% of expert, intermediate and novice strongly agreed that CoMADE supports the design of polymorphic task model. 85.71%, 53.85%, 12.50% of expert, intermediate and

Novice agreed that CoMADE supports the design of polymorphic task. Similarly, 15.38%, 12.50% of intermediate and novice could not decide.

Fig 8 shows that 14.29%, 15.38%, 25.00% of the expert strongly agreed that CoMADE supports dynamic creation and persistence of CIO variant implementation details. 57.14%, 61.54%, 50.00% of expert, intermediate and novice agreed that CoMADE supports dynamic creation and persistence of CIO variant implementation details. 28.57, 23.08%, 25.00% of expert, intermediate and novice could not decide.

Fig. 8 shows that 43%, 54%, and 63% of expert, intermediate and novice strongly agreed that CoMADE generates user interfaces according to the specified user preferences and needs. Similarly, 57%, 38% and 25% of expert, intermediate and novice agreed that CoMADE generates user interfaces according to the specified user preferences and needs, but 8% and 12% of intermediate and novice could not decide.

VI. CONCLUSION

We have developed a model based toolkit named CoMADE for dynamic generation of user centric interfaces for J2ME enabled handheld devices. CoMADE generates user centric interfaces based on real-time or pre-recorded user information. Emulator programs are integrated with CoMADE to achieve design time evaluation of generated interfaces. The toolkit allows designer to modify an interface artifact and extend an application with ease by a drop of additional variant into the variant pool. CoMADE achieves direct user participation in user interface generation with a high degree of dynamism and flexibility.

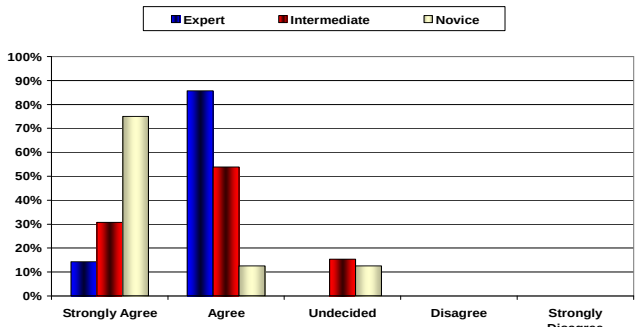

Fig.7 CoMADE Supports the Design the of Polymorphic Task Model

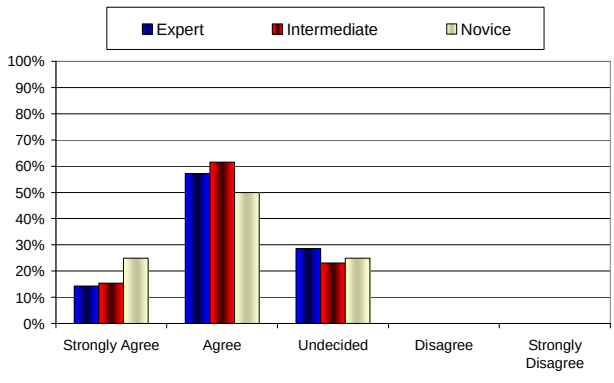

Fig. 8 CoMADE Supports Dynamic Generation and Persistence of CIO variant implementation details

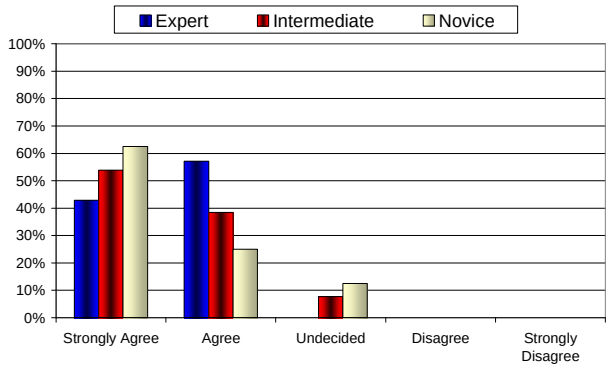

Fig. 9 CoMADE Generated User Interfaces According to Specified Preferences and needs

References

[1] Vasilis Koudounas, Omar Iqbal, (1996) “Mobile Computing: Past, Present and Future”: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/vk5/report.html

[2] Franck Tétard4, Erkki Patokorpi4 and Vaida Kadytė4, (2005) "User-Centred Design of Mobile Services for Tourists.A Case Study on Student Work on Mobile Design" IFIP International Federation for Information Processing.Vol. 158/2005,pg. 155-168,

[3] Ipadeola A.O, Iyilade J.S, Adigun M.O, Xulu S.S (2008) “A multi-target User Interface Design For Pervasive Grid Environment”. Accepted For Publication.

[4] Tao Guan, Ed Zaluska, David De Roure; (2005).” A Grid Service Infrastructure for mobile Devices”. In proceedings of 1st Semantic Knowledge and Grid conference, Beijing, China

[5] Justyna Burns (2001) Gregory R Madey "A Framework for Effective User Interface Design for Web-Based Electronic Commerce Applications" International Journal of an Emerging Tran discipline, vol. 4, pp 67-75.

[6] Aaron Marcus, (2003) “Universal, ubiquitous, user-interface design for the disabled and elderly” Interactions, v.10 n.2.

[7] Aaron Marcus, . (2002) “Dare We Define User-Interface Design?” Interactions, v.10 n.2.

[8] Olsen, D., Jefferies, S., Nielsen, T., Moyes, W., Fredrickson, P., (2000), ”Cross-modal Interaction using XWeb”. In Proceedings of the UIST’2000, CA, USA

[9] kai Richter; A Transformational Approach to Multi-Device Interfacea” (2005), In proceedings of human Factor in computer System, Portand oregon, USA.

[10] Barbara b,. (2007) “Designing Mobile User Experience”. John Wiley & Sons, ISBN: 978-0-470-03361-6.

[11] David Thevenin, Joelle Coutaz, Gaelle Calvary; (2001). A Reference Framework For the Development of Plastic User Interfaces; Proceedings of Engineering of Human Computer Interaction Conference (EHCI')

[12] Paterno, F., (2005). Model-Based Tools for Pervasive Usability. Interacting with Computers, vol. 17, issue 3, pp. 219-315

[13] James Lin, Damask: (2003) A Tool for Early-Stage Design and Prototyping of Cross-Device User Interfaces. In Confer ence Supplement of UIST 2003: ACM Symposium on User Interface Software and Technology, Vancouver, BC,Canada, , pp. 13–16

[14] C. Wiecha et al., (1990) "ITS: A Tool for Rapidly Developing Interactive Applications," ACM Trans. Information Systems, pp. 204-236.

[15] Lonczewski, F. and Shreifer, S., (1990) The FUSE System: an Integrated User Interface Design Environment, In Proceedings of the working conference on Advanced visual interfaces: CADUI'96, pg, 37-56.

[16] Angel R. Puerta, "A Model-Based Interface Development Environment," IEEE Software, vol. 14, no. 4, pp. 40-47, Jul/Aug, 1997

[17] Annett, J., & Duncan, K. D. (1967). Task analysis and training design. Journal of Occupational Psychology, 41, 211-221.

[18] Germanakos P., Samaras G., Mourlas C., & Christodoulou E. (2006) " Innovative Personalization Issues for Providing User-Centric mGovernment Services ", Proceedings of the Second European Conference on Mobile Government (Euro mGov 2006), Brighton, August 30-31 & September 1, 2006, pp. 124-133.

[19] Schlungbaum, Egbert. “Individual User Interfaces and Model Based User Interface Software Tools”. Poster Session, 1997 International Conference on Intelligent User Interfaces.

[20] Dietrich, H., et.al. (1993). State of the Art in Adaptive UserInterfaces. In [15], 3-48.

[21] Tim Fielden, Ana Orubeondo , (2000) "WAP vs. J2ME: Which is best for mobile?" http://www.infoworld .com/articles/tc/ml/00/11/27/001127tcpcp.htmlInfoWorld TestCenter

[22] Gupta A, Srivastava M. “Integrated Java Technology for End-to-End m-Commerce”, (2001). Available Online :http://developers.sun.com/mobility/midp/article/mcommerce/

[23] Ipadeola A, Olugbara O, Xulu, S, Adigun M, “Polymorphic Logical Description for Automatic Generation of User Centric, Multi-Device Interfaces”. Submitted for Publication.

[24] Savidis A., Paramythis A., Akoumianakis D., Stephanidis C. (1997).“Designing user-adapted interfaces: the unified design method for transformable interactions”, In Proceedings of the conference on designing interactive systems: processes, practices, and techniques,

[25] Fabio Paterno ; Multimodality and Multi-device interfaces; In: W3C Workshop on Multimodal. Interaction.Sophia Antipolis, France. 2004. 17.Pleuß, A.

[26] Grehan R.” Complex Object Structures, Persistence, and db4o “, White Paper On Open Source Object Database in Java and .NET. Available Online: http: //www.db4o. com /about /product information /whitepapers/db4o %20 White paper %20- %20Complex%20Object%20Structures.pdf

[27] Nielsen J. “Why You Only Need to Test with 5 Users” Available Online: http://www.useit.com /alertbox /20000319.html.

Mr A.O Ipadeola is with the University of Zululand where he acts as a research assistant in the department of computer science. He is currently pursuing a Master of Science degree in Computer Science. Mr Ipadeola obtained his first Bachelor of Science (Honors) degree in computer science at University of Ilorin in Nigeria. His research interests include: pervasive grid computing and artificial intelligence. (email:profyommexy@yahoo.ca, +27726238358).

Mr O.O Olugbara obtained his Bachelor of Science (Honors) degree in Mathematics at the University of Ilorin, Nigeria. His Master of Science was in Computer Science at the same institution. He worked as a senior lecturer both in University of Ilorin and Covenant University in Nigeria. He is currently pursuing his Doctorial degree in the department of Computer Science in University of Zululand. His research interests include Software Engineering and Artificial Intelligence.(email:oluolugbara@gmail.com, +27780428567)

Prof. M.O. Adigun is a full time staff member and head for both the department of Computer Science and the Centre of mobile eServices at the University of Zululand. His research interests include: Software Engineering and Grid Computing.(email:profmatthewo@gmail.com, (+27845138232)

Dr. S.S Xulu is a full time staff member and a senior lecturer in Computer Science department at the University of Zululand. His research interests include: Software Engineering and Grid Computing.(email:ssxulu@gmail.com, +27832276987)