MobiNET: A Framework for Supporting Java Mobile Application Developers through Contextual Inquiry

Ogunleye O. Samuel
African Advanced Institute for Information Technology, Meraka Institute, Council For Scientific Industrial Research, Meiring Naudé Road, Brummeria, Pretoria, South Africa

Abstract—From a developing world perspective, mobile phone is the primary technology for the majority of people and will be for the foreseeable future. It will be their connection to the internet, their communication tool, school book, vaccination report, photo album and many other things. Despite this reality, relatively few mobile applications exist. However, a tool to support the programming of mobile applications can significantly impact and improve programmer's productivity and software quality. In this paper, we develop a concept for a mobile tooling framework that extends the Netbeans integrated development environment (IDE) for mobile programming. Mobile Tools for Netbeans (MobiNET) design would support the development of mobile applications for various mobile phones. We evaluate Netbeans as a development tool and drawing from insights gained in interviews with mobile software developers, conceptualize MobiNET. A working prototype is discussed and evaluated.

Keywords— Integrated Development Environment, Mobile Applications, MobiNET, mobile tools.

I. INTRODUCTION

One Vision of mobile computing is to deliver powerful applications through devices one can easily carry. To achieve this vision, mobile computing application development requires collaborations between mobile applications and the development environment (i.e. the IDE). However, tools support for mobile applications development can significantly improve programmers' productivity and the quality of mobile applications for various Java mobile phones. Integrated Development Environments (IDEs) – such as NetBeans, Eclipse, JBuilder, etc. are tools of choice for developing Java mobile applications and they are also instrumental in developing individual components for mobile applications [11].

Developing mobile applications using any of these development environments is a complex task [11]. However, one vision of mobile applications developers is to deliver robust and comprehensive applications for various Java enabled mobile devices that one can easily carry, through using one of the IDEs mentioned above.

The popularity of mobile applications and services are now such that we feel it is time to look at how well mobile applications developers are supported through these existing development tools.

This research uses Contextual Inquiry (CI) [3] to investigate how mobile applications developers can be supported through Java IDEs in order to identify problems that are encountered when using Java IDEs to develop mobile applications. We also assess the utility of CI for extracting the design requirements for the IDEs. NetBeans IDE was considered as the ideal IDE to use for this research. This is due to the fact that it is an open source IDE and it is considered as the most widely used Java IDE for developing mobile applications for mobile devices [1].

Therefore, we were interested in finding ways to improve the usability of Java IDEs for mobile applications development and to provide more support for Java mobile applications developers through the IDEs.

The techniques of CI recommended observing activities as they occur in their natural context in order to be able to portray the process of the work as well as the discovery of the places where technology could be applied to defeat the observed difficulties [5]. This method was chosen because it would provide data about the detailed problems faced by Java mobile applications developers when using a Java IDE to develop mobile applications and it also provide guidance on the design of the supporting framework [4], [9].

II. CONTEXTUAL INQUIRY

CI, as described by Beyer and Holtzblatt is a structured approach to the collection and understanding of data from fieldwork with the purpose of building a tool that supports the user of a system. It is a method that provides the researcher and/or designer with a grounded and detailed knowledge of users' work as a basis for their design [14]. This is usually achieved by fostering a strong relationship with the users. This will determine how well the researcher/designer understands the users in order to be able to support them. And users are always assumed as the expert in their work [3], [4].

CI is always achieved through a face – to – face interaction using an apprenticeship model which provides an attitude of inquiry and learning while the users are being studied [3], [4] and it defines a clear set of concerns rather than a list of specific questions which enables the researcher/designer to focus on a few key issues and gather concrete data during the session that they may have with the users.

The importance of CI is that you can ask questions and prompt for explanations immediately [3], [9].

978-1-4244-3523-4/09/$25.00©2009 IEEE

Goal	Run code for various devices
Content	Code was working fine but could not be adapted to different mobile devices without changing the code
Outcome	While trying to make sure that the code works fine on all the devices, there are various versions of codes of emanated from only one written codes.

Table 1: The Table showing the action that users wish to achieve.

III. Methodology

Using CI, Sixty four expert mobile applications developers were recruited and observed in their place of work, that is, the Computer Science department laboratory of the University of Cape Town. This was done during the first and second semester of the session. The researcher met with each mobile applications developer and explained the motivation behind CI that is to identify programming difficulties that mobile applications developers experience while developing mobile applications using one IDE or the other in order to be able to provide support for these difficulties. This would be achieved by observing the developers as they develop their mobile applications using various Java IDEs However, as developers develop their applications, the researcher recorded the observations on both the paper and video.

Hypothesis about the programmer actions were formed. This was later shown to the developers. For example, "You want to be able to port your applications to various mobile devices by using the functionalities provided by the IDE." and the developers would reply, "Actually, I want to be able to write one set of code and by using this IDE, I want my application to work on different mobile devices without changing the codes." This is represented in Table 1 above. Participants were however, paid approximately $10 for their time and participation.

IV. Findings and Results

Understanding how and where to improve the environment for Java mobile developers, working on Java mobile applications requires some investigation in order to learn how they do their work, while using a particular Integrated Development Environment (IDE) for developing mobile application [11]. In order to achieve this, the researcher/designer has to conduct an observation of user. However, this can be achieved through using contextual Inquiry [3].

To this end, we conducted an observation of Sixty four mobile application developers while they were using their various IDEs to develop mobile application using contextual Inquiry (CI).

The following data and results were collected and obtained from our CI method: 84% felt that NetBeans supports the way in which they work. However, after further use of CI as well as follow-up interview [4], [9], it was clear that mobile developers expect mobile application to run correctly on all Java 2 Micro Edition, J2ME-enabled software and hardware (e.g. J2ME-Enabled mobile devices). But this is not always the case [6].

Hence the result from our method of CI showed that in a typical development, porting and testing Java mobile applications using a Java IDE such as NetBeans, Eclipse, JBuilder, etc. take longer time than expected in order to be able to accommodate the varieties of devices to be supported through a particular IDE.

V. Design (Supporting Mobile Developers)

After we analyzed the results, our research then focused on better supporting mobile applications developers while developing mobile applications for a variety of mobile device platforms. This however, can be done through a development environment (that is, the IDE), since almost all mobile applications developers are now developing mobile applications through one IDE or the other [11].

However, our research focused on the NetBeans IDE. This is because the NetBeans IDE is an open source environment which allows for alterations and also, the result from our research in addition with the literature survey conducted showed that, NetBeans is considered to be the most widely used IDE for developing Java mobile applications [1].

Therefore, we designed a plugin to be incorporated into the NetBeans IDE. The plugin we built is called Mobile Tools for NetBeans (MobiNET) which can be used to aid the development of mobile applications that can be easily ported into different mobile devices through using NetBeans IDE without the need to adapt the application for each mobile device profile. MobiNET's major function is to help mobile developers preprocess source code to adapt mobile applications to various mobile devices. The goal is to keep only one form of source code which, when preprocessed, generates code and metadata which can be executed correctly on J2ME-enabled devices. The source code only needs to be written once along with accompanying directives for the tools. A device database, which is an XML file, only needs to be altered to contain all the devices the programmer wishes to target. Tool directives are used to preprocess the source code. These are simple code snippets that help during the preprocessing stage. All the directives start with the Java comment code (that is the two forward slash //) followed by the pound symbol (#).

VI. Evaluation

The results from the contextual inquiry have been applied in implementing a system to support mobile developers as presented in the design section. However, [4] argued that designers may not know how useful their system is until an evaluation has been carried out. Hence, the evaluation of our system.

Our evaluation focused on determining the tasks the users achieved in using the system, rather than evaluating the system performance [13]. Also we were not so interested in how efficient the users are in using the design but rather how well the system supports the goal of the user [9]. Therefore, a prototype application has been developed for the purpose of this evaluation. This is because we were not interested in knowing whether a mobile application programmer knows how to write code but rather how well the system can help

the programmer achieve the task for which it was designed [2].

To this end, a sample application was developed for the evaluation purpose and this was a simple mobile menu. This is a simple application and was developed because we wanted the tasks that would be carried out by users to be simple enough so that users will be able to evaluate the system successfully [2], [9].

A. Subject in the Evaluation

During the evaluation of a system, it is imperative to choose subjects who currently use, or will use the design [2], [8]. However, [9] argued that when conducting evaluation, it is important to recruit subjects who represent the sample population for which the system is targeted e.g. users with some range of expertise in the context of the study. In this study, the subjects were those who have had experience in developing mobile application.

Also, it will take many more than five users to successfully evaluate a system [12], [10] also suggested that more than five or seven subjects per cell is the recommendation for the evaluation of a system or design where a cell represents a class of subjects who represents the users. Therefore, we evaluated our design with seventeen (17) subjects, all of which were students from Computer Science (3 PhD students, 7 Masters Students, 5 Honours students and 2 Undergraduate students).

B. Task in the Evaluation

The following three tasks were developed in order to evaluate the MTN that was developed to support mobile applications developers.

Task 1: To develop a simple mobile application and preprocess it according to the various devices of their choice based on the experience acquired in the tutorial.

Task 2: To write a build (XML) file based on the experience acquired during the tutorial session.

Task 3: To use the build file to build and preprocess the application to various devices as defined in the device collections.

The chosen topics for the tasks were indentified to be simple to use during the evaluation after [8] suggestion on tasks to be used during evaluation and therefore were considered most important. The efficacy of the tasks was also reviewed by colleagues as well as the consulting HCI expert during the design of the questionnaire. A pilot study was also conducted with the potential users who would not be involved in the main evaluation study in order to determine the viability of the experimental procedure [9]. This also helped us to decide the criteria for what would constitute successful completion of the task.

C. Evaluation Environment

In order to guarantee comfort and provide a familiar environment, the evaluation was conducted in the postgraduate computer laboratory of the Department of Computer Science at the University of Cape Town. The users' privacy and their confidentiality were maintained throughout the process of the evaluation. This, we did in order to consider the ethical issues that are related to user evaluation as pointed out by [9].

D. Evaluation Procedure

On arrival, the agreement/consent form was given to subjects to fill, sign and submit,. After this, subjects were introduced to the system as well as the evaluation that was to be performed and instruction on how this would be done was given. The purpose of this was to make sure that all subjects were given the same information and instruction.

The subjects were asked to sit alone with a computer system running Windows XP and NetBeans version 5.5 as well as Java Development Kit (JDK) 1.5. Each subject that participated in the evaluation study did so seperately. Before starting the main tasks, subjects were given a copy of the sample mobile menu application and a sample of the build.xml file that would be used to run the application and were instructed to explore the sample application for up to 10 to 15 minutes to familiarize themselves with it.

Each subject was then asked to walk through the three tasks and they were asked to tell us what they were thinking as they walk through the samples and as they perform the tasks (think aloud) [4], [9]. They were given up to 10 minutes for the first task, 20 minutes for the second task and 10 minutes for the third task. If they did not finish a task within the allotted time they were ask to stop. When all the tasks were completed, the subjects were given a post-test questionnaire which consists of items derived from the [QUIS user satisfaction questionnaire] to fill and returned before leaving the evaluation room. When the questionnaire was completed, a debriefing session and an unstructured interview were held in which the subjects were asked for their opinion [9].

We wanted subjects to complete these tasks to investigate and assess the suitability of the application as realistically as possible based on the following three assessments:

- How well the application was designed.
- How easy the system was, in terms of time to complete tasks by subjects and error rates during task completion.
- How well the system supports mobile developers in developing applications for specific devices.

In summary, there were four different sections during each evaluation and all these took up to 1 hour on the average. These sessions were:

- Introduction of the system and the experiment to perform
- Tutorial about the system
- Carrying out tasks using the system
- Questionnaire administration, debriefing session and unstructured interview.

E. Results and discussion

Responsiveness is the most important factor in determining users' satisfaction with a system [9]. All the users found the system satisfying.Users liked the fact that little needed to be done when using the tool as they only need to perform some changes in the configuration file. This was further confirmed in the unstructured interviews conducted after the evaluation.

The result of our observation coupled with the users response from the questionnaire shows that 78% of the subjects find it simple to quickly learn how to operate the system while 72% of the subjects got started with the system quickly. It was observed that only 1% of the users found it a little difficult to get the scope of the system at the beginning.

The result from the users' response showed that the time to learn and operate the system was very quick. Also, our result showed that 78% of the subjects agreed that the system was very fast; it took less than 10 seconds to preprocess an application for 30 different mobile devices.

Our result also showed that 76% of the subjects agreed that the system was reliable because when using the system, no error was encountered. This is because the errors have been pointed out and dealt with, during the pilot study. However, 75% of the subjects agreed that the ease of operating the system depends on the level of experience that a subject has in programming Java mobile applications. These results are represented in figure1.

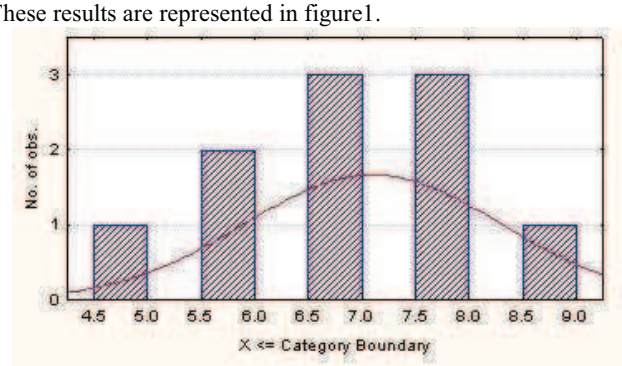

Figure 1. Graph result for Overall user reaction to the System

VII. CONCLUSION

Through this research, we have been able to establish that contextual inquiry, which forms part of the new generation observation methodology, is the best to improve the usability of a system. This is because it allows a researcher to learn more about the users' activity in order to be able to provide support for them.

However, the major goal of this research study was to establish how we can support mobile application developers through a Java IDE by using contextual inquiry. NetBeans was the IDE of choice for the purpose of this research. As stated earlier, this was because NetBeans was considered as the most widely used IDE for Java mobile application development [1]. A mobile tool called Mobile Tools for NetBeans (MobiNET) has been designed and developed to support our research. A set of different configuration descriptions for mobile devices was designed, implemented and were put together to form the MobiNET. We conducted the evaluation of the tools to establish whether the tool presented a more effective, efficient, and satisfying solution than those currently available.

The research that was conducted and presented in this paper provides support for Java mobile application developers through a Java IDE. It is our belief that more researchers and designers should use the ideas presented in this paper to support their intended users.

ACKNOWLEDGMENT

The author would like to thank Merryl Ford, JP Tolmay, Louis Coetzee, Pieter Van zyl, Hinal Patel, Ilse Viviers, Adele Botha, Sokoya Dare, all of the Meraka Institute. Also, the author will like to give a special thank goes to Ogunleye Tozama for her support and understanding through this research work.

REFERENCES

[1] Benson, C., Muller-Prove, M., and Mzourek, J. (2004). Professional usability in open source projects: GNOME, OpenOffice.org, Netbeans. *Proc CHI 2004* ACM Press (2004), 1083 – 1084.

[2] Dumas, J.S. and Redish, J.C. (1999). *A practical guide to usability testing.* Great Britain:: Cromwell Press.

[3] Holtzblatt, K. and Beyer, H. (1997). Contextual Design: Using Customer Work Models to Drive Systems Design. *Ext. Abstracts CHI '97.* ACM Press (1997), 184-185. Atlanta, USA: ACM

[4] Jones, M., and Marsden, G., (2005). *Mobile Interaction Design. England*. Wiley.

[5] Cross, K., and Warmack, A. (2000). Contextual Inquiry: Quantification and Use in Videotaped Analysis. *CHI (2000)*, ACM Press (2000), pp. 317-318.

[6] Mace, M. (2006). *Mobile Opportunity*. http://mobileopportunity.blogspot.com/2006/05/flash-versus-windows-can-adobe-break.html.

[7] Nielsen, J. (1996). *International usability testing. In E. del Galdo and J. Nielsen (Eds), International user interfaces.* New York: John Wiley and Sons.

[8] Nielsen, J. (2003., January,). *Recruiting test participants for usability studies*. http://www.useit.com/alertbox/20030120.html.

[9] Preece, J., Rogers, Y., and Sharp, H. (2007). *Interaction Design – Beyond Human – Computer Interaction.* West Sussex: John Wiley & Sons Ltd.

[10] Scholtz, J. (2004). *Usability Evaluation*. *Publication No. 545.* National Institute of National Institute of Standards and Technology http://www.itl.nist.gov/iad/IADpapers/2004/Usability%20Evaluation_rev1.pdf

[11] Soroka, D., Cáceres, R., Dig, D., Schade, A., Spraragen, S., and Tiwari, A. (2006). Pegboard:: A framwork for developing mobile applications. *Proc. MobiSys '06.* ACM Press (2006), pp. 138-150.

[12] Spool, J. and Schroeder, W. (2001). Testing web sites: Five users is nowhere near enough. *Ext Abstracts CHI 2001.* ACM press (2001), 285-286.

[13] Thomas, K. (1999). Designing a Task-Based Evaluation Methodology for a Spoken Machine Translation System. *Proc. 37th ACLCL.* ACLCL Press (1999), pp 569-572.

[14] Wixon, D., and Raven, M., E. (1994). Contextual Inquiry: Grounding your design in user's work. *Proc. CHI '94.* ACM Press (1994), pp 409-410.