

# Model-Driven Development of Mobile Applications

Florence T. Balagtas-Fernandez

Adviser: Heinrich Hussmann

Department of Computer Science, University of Munich, Germany

florence.balagtas@ifi.lmu.de, hussmann@ifi.lmu.de

<http://www.medien.ifi.lmu.de>

## 1. Problem Statement

This research aims to simplify the creation of applications for mobile platforms by developing a high-level and platform independent model of an application, and automatically transforming this high-level model to platform specific code. The research method is a combination of the model-driven development (MDD) approach in software development and application of techniques in the field of human-computer interaction (HCI) particularly on user-centered system design.

This research involves developing a graphical modeling language which is specific to mobile applications, and coming up with a generic algorithm for the conversion of this graphical model into code. The main focus of the research however, will be on the design of the graphical model, and the interaction techniques which will allow non-expert people<sup>1</sup> to create specialized mobile applications with ease. Key research questions that need to be answered are: Which level of abstraction and modeling constructs are adequate for non-expert people to create specialized mobile applications? How can technical details such as device limitations, transmission of information to other devices or to the network, etc., be abstracted from the non-expert user as they model the application so that they can focus more on the design and logic of their application, but at the same time take advantage of advance capabilities of mobile devices (GPS, Bluetooth, Wifi, etc.)? How can the design of interaction with other devices that connects to mobile applications become simpler? Do non-experts want the modeler interface to look like the actual application or do some high level form just like in UML would be enough (example: property editors, flow arrows)?

More details on the proposed methodology of this research are discussed in section 3.

---

<sup>1</sup>Non-expert people here are defined as those who have no experience in developing applications for mobile platforms

## 2. Review of Related Literature

Mobile devices are becoming the most indispensable computing device a person needs. With the advent of smart phones, the mobile phone has become as powerful as any ordinary desktop computer in terms of the applications it can run and its computing power [5]. People are now using their mobile phones not just for calling and sending messages but also for browsing the Internet, playing games, listening to music, taking pictures and videos just to name a few. In response to the rising demand for better mobile devices, companies have developed various types of mobile devices that would suit different people with different computing needs. Some of these companies are even opening up their application programming interfaces (APIs) that would allow anyone to develop their own applications for their mobile devices given the right tools. Examples of these are the Symbian operating system C++ API<sup>2</sup>, Java platform micro edition (Java ME) API<sup>3</sup> from Sun Microsystems, the Android API<sup>4</sup> from the Open Handset Alliance and the iPhone API from Apple<sup>5</sup>.

One problem about this though is that, developing any type of computer application, whether it is for the desktop computer or mobile device is quite difficult. It still takes a large amount of skill and familiarity with how the framework is used before a person can create a decent amount of code for a simple application. Even setting up the programming environment is a complex task, let alone, trying to figure out how to use the APIs, compiling, running and deploying the application on the actual device. Other things that makes developing applications for mobile devices more difficult as compared to desktop applications are factors such as device limitations (e.g. screen size, computing power, power consumption) [12], different operating systems for mobile devices, different data representation and additional

---

<sup>2</sup><http://www.symbian.com/>

<sup>3</sup><http://java.sun.com/javame/index.jsp>

<sup>4</sup><http://code.google.com/android/>

<sup>5</sup><http://developer.apple.com/iphone/>

device capabilities (e.g. Bluetooth, Wifi, GPS, Camera-enabled) which are not standard to all devices and therefore should be considered when developing a uniform application that can be run on different mobile devices.

An alternative approach to solving interoperability and development problems for mobile platforms is through creating web-based applications [12]. The problem with this though is that, web-based applications are limited to business processes and is not applicable to applications that need to access additional device capabilities such as built-in Camera, GPS and Bluetooth to name a few. Particularly in the area of pervasive healthcare [17], applications such as remote health monitoring, location tracking and those that need to connect to body sensors still need full device access which is only possible at the moment through native mobile applications.

Going back to the problem of complexity of developing software, one solution to try to ease the application development process is by providing Integrated Development Environments (IDEs), wizards and tutorials. For mobile phone development in particular, there are several IDEs available for developing on a Symbian phone like Code Warrior from Metrowerks, C++ Builder Mobile Set from Borland and Visual Studio from Microsoft [10]. The Netbeans Mobility Pack is also available for developing J2ME applications [2]. The Android platform provides a development environment, a device emulator, debugging tools and plugins for the Eclipse IDE [1]. Android also has some tutorials on their website on how to get started and implement various applications for Android supported devices. However, the solution of IDEs for ease of application development does not address the problem of developing applications that runs on different devices. This is especially true for applications that access specialized device capabilities such as Bluetooth or device database. Common approach to development is creating applications separately that would run on the different devices, but with the same design and application logic. This makes the task tedious and redundant for programmers, especially if there are changes to application specifications.

This research proposes applying model-driven development (MDD) specifically to the development of software applications running on mobile platforms. Model-driven development (MDD) is an approach to creating complex software systems by first creating a high-level, platform-independent model of the system, and then generating a specific code based on the model to the target platform [13]. MDD would greatly help in easing the development process of application development for mobile platforms since it would address the problem of redundancy of tasks in developing an application that would run on several platforms. By having a high-level model that would describe the application and having this model be transformed to platform

specific code for different mobile platforms, the programmer's task of rewriting code over and over again is greatly reduced.

Some other researches which applies MDD to ease software development are: the Simple Mobile Services project<sup>6</sup> [6][3][11] which applies MDD to the creation of mobile services, PervML [14] which aims to create pervasive systems through the application of MDD, the Multimedia Modeling Language (MML) [15] which is designed for interactive multimedia applications, and researches like [9] and [7] which applies modeling concepts for creating platform independent user interfaces. The examples stated encompass different types of applications ranging from mobile services, multimedia applications and platform-independent user-interfaces which clearly shows that model-driven development is a promising approach in aiding software development.

The Model Driven Architecture (MDA) proposed by the Object Management Group focuses on the creation of a Platform Independent Model (PIM) which is automatically converted by transformation tools to Platform Specific Model(s) (PSM), which are then transformed to specific codes [13]. An example of a PIM that can be used is the Unified Model Language (UML). A problem with this approach is that using UML to model a specific kind of problem complicates the problem solving process even more especially if the user is not that familiar with the UML constructs. The current UML tools that claims to transform the model to code are only specific to general purpose programming languages like C and Java, and no support yet for mobile platforms. We propose to find an alternative way of modeling mobile applications that particularly addresses the factors to be considered when modeling applications for mobile devices. The modeling constructs should be more intuitive and easy to understand, that no specialized knowledge is necessary to use it. This is where the user-centered approach comes in. We want to apply the concepts of MDD in the creation of mobile applications, however, we also want the modeling constructs to be easy to understand and intuitive enough that even non-experts will be able to design their own applications with ease.

### 3. Proposed Research Method

This research aims to simplify the creation of applications for mobile platforms by applying model-driven approach to software development combined with techniques in user-centered design in the field of human-computer interaction.

In order to apply and test the design concepts, a modeling tool called Mobile Applications (MobiA) modeler will be created. Several mockups of the tool will be created and

<sup>6</sup><http://www.ist-sms.org/>

tested with a group of non-expert users. The following is a summary of the basic steps that would be done in order to accomplish this research.

1. For the theoretical part, research will be done on the areas of mobile applications development, end-user computing, user programming and model-driven development.
2. For the practical part of the research, user surveys will be conducted in order to know the users' background knowledge in programming and what they wish to have for a development tool based on their previous experiences if there are any.
3. Create an initial prototype based on existing approaches with innovations based on user surveys and other researches.
4. Test the initial prototype by letting target users use the tool and observing which parts they have difficulties with.
5. Modify the prototype based on the result of the initial prototype testing.
6. Come up with a generic algorithm to convert the graphical model into code.
7. Iterate steps 4 and 5.

**The Target Users.** The target users of this system are the non-expert people who have no background in creating applications for mobile platforms. In analogy to MDA, these are the PIM analysts who know the functionality and requirements of a target application that they want to run on mobile devices. The challenge is how to provide support to the non-expert users such that it is easy for them to model the application that they want without having knowledge in programming. In these cases, the modeler is actually also the end-user who will be using the application for his/her needs. Examples of such non-expert users are researchers who need mobile applications that they will use for their research studies on the field. These applications are very specialized that it is difficult for them to find a generic application. What happens usually is that they hire some programmers, tell the programmers what they need and the programmers create the applications for them. One example research project which does this is the Worldwide experimental Platform (WeP<sup>7</sup>) project from the LMU Institute of Medical Psychology. The WeP aims to utilize the Internet in order to conduct studies in the fields of medicine, epidemiology and genetics [4]. One of their researches involves an application<sup>8</sup> that runs on a personal digital assistant that executes psychophysical tests, psychomotor vigilance tests and other types of tests in order to get information from test subjects. These types of applications are so specialized that they have to let some programmer create the mobile application for them which would run on that specific

<sup>7</sup><http://www.thewep.org>

<sup>8</sup>Information regarding the application was collected during an interview with one of the researchers involved in the project

device. If we apply MDD to the creation of mobile applications, these medical researchers would just have to model what type of application they want and then let the modeling application capable of code transformations create the mobile application that would run on that device. Applications such MATLAB<sup>9</sup> and STELLA<sup>10</sup> have been helping scientists and researchers through the years, but with the advent of mobile computing, these scientists need to benefit from the new technologies that would allow them to do their researches and experiments on the go. And instead of relying on programmers for creating these specialized applications, it would be easier and at the same time cost-efficient for them to create their own applications with the aid of these modeling tools.

**From Graphical Model to Code** Although transformation of the graphical model to code is not the main focus of this research, further research on the appropriate transformation tools will be done for proof of concept. In this step, the high-level graphical model created from the MobiA modeling tool would then be transformed to a more formal model such as an XML-based model that can be parsed and processed in order to generate the platform specific code.

**Expected Output** The expected output of this research is a list of guidelines for modeling mobile applications for non-expert users and as a proof of concept, the MobiA modeler prototype that illustrates the modeling constructs for the graphical model which will allow non-expert users to model their applications and produce code that could run on mobile platforms.

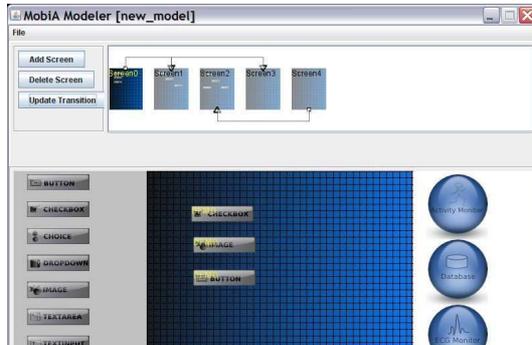
## 4. Current Status

Initial surveys have been done to non-expert users which is a group of researchers who are part of the WeP project. These people are willing to create their own mobile applications to be used for their own experiments and researches. The survey involved asking the participants about some personal information, their background in computers, some information about the WeP and their wishes for a development tool. The results differ for every person in a way that they all have different specializations, technical backgrounds and used different types of operating systems. With regards to their wishes for the tool, most of them would like the tool to be graphical in nature and feature a drag-and-drop environment.

<sup>9</sup><http://www.mathworks.com/products/matlab/>

<sup>10</sup>STELLA-Systems Thinking for Education and Research, <http://www.iseesystems.com/software/Education/StellaSoftware.aspx>

An initial prototype for the MobiA tool has been done as shown in figure 1. This prototype shows the different components that a typical application contains. These includes the user interface model which shows the user interface for a certain application, the navigation model which shows how an application navigates from one screen to the next and the information retrieval model which aids in showing how information is passed from one component<sup>11</sup> to another.



**Figure 1. Initial Prototype**

This initial prototype is important in visualizing which components will be important in developing mobile applications. Later on, different variants of this MobiA application will be created and tested with non-expert users. The different mockups of the tools will then be compared to each other and to existing tools.

## References

- [1] Android documentation, <http://code.google.com/android/documentation.html>.
- [2] Netbeans, <http://www.netbeans.org/kb/articles/mobility.html>.
- [3] The sms project, <http://www.ist-sms.org/>.
- [4] The wep, <http://thewep.org/>.
- [5] G. D. Abowd, L. Iftode, and H. Mitchell. Guest editors' introduction: The smart phone—a first platform for pervasive computing. *IEEE Pervasive Computing*, 4(2):18–19, 2005.
- [6] G. Bartolomeo, N. B. Melazzi, G. Cortese, A. Friday, G. Prezerakos, and R. W. S. Salsano. Sms: Simplifying mobile services - for users and service providers. In *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*. IEEE Computer Society, 2006.
- [7] A. Coyette, S. Kieffer, and J. Vanderdonckt. Multi-fidelity prototyping of user interfaces. In *Human-Computer Interaction INTERACT 2007*, volume 4662/2007, pages 150–164, Berlin / Heidelberg, 2007. Springer.

<sup>11</sup>Component here pertains to either a UI element, information source or storage (e.g. database, Internet) or any other type of tangible device that provides information (e.g. sensors, other mobile devices)

- [8] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction*. Prentice Hall International, 1993.
- [9] J. Eisenstein, J. Vanderdonckt, and A. Puerta. Applying model-based techniques to the development of uis for mobile computers. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 69–76, New York, NY, USA, 2001. ACM.
- [10] R. H. et al. *Symbian OS C++ for Mobile Phones*. John Wiley and Sons Ltd, West Sussex PO19 8SQ England, 2003.
- [11] I. Foukarakis, G. Kapitsaki, D. Kateros, G. N. Prezerakos, E. Kosmatos, R. Vrnieni, G. Broll, M. Maurer, D. Andreansky, A. Friday, M. Harding, and O. Storz. Sms project deliverable 4.3.1-design of service authoring wizard. Technical report, 2007.
- [12] M. Gaedke, M. Beigl, H.-W. Gellersen, and C. Segor. Web content delivery to heterogeneous mobile platforms. In *ER '98: Proceedings of the Workshops on Data Warehousing and Data Mining*, pages 205–217, London, UK, 1999. Springer-Verlag.
- [13] A. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Pearson Education, Inc., Boston, USA, 2003.
- [14] J. Munoz, V. Pelechano, and J. Fons. Model driven development of pervasive systems. In *European Research Consortium for Informatics and Mathematics (ERCIM) News*, volume 58, pages 50–51, 2004.
- [15] A. Pleuss. Mml: A language for modeling interactive multimedia applications. In *ISM '05: Proceedings of the Seventh IEEE International Symposium on Multimedia*, pages 465–473, Washington, DC, USA, 2005. IEEE Computer Society.
- [16] J. Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2006.
- [17] U. Varshney. Pervasive healthcare and wireless health monitoring. *Mob. Netw. Appl.*, 12(2-3):113–127, 2007.