Design a medical application for Android platform using model-driven development approach

J. Yepes¹, L. Cobaleda², J. Villa D¹, J. Aedo¹
¹ARTICA, Microelectronic and Control Research Group
²ARTICA, Software Engineering Research Group
University of Antioquia
Medellín, Colombia

Abstract - Since the complexity of embedded systems has grown significantly, it has been necessary increase the abstraction level. For that reason, we propose a method to design a system for medical device interoperability, called SIMMIT (System Integration Medical Monitoring and Interoperability for Tele-care) based on a Model-Driven Development approach. This work presents a strategy and the tools to design a software application for embedded systems supported by functional and non-functional requirements. This approach starts with a system specification report, which describes both the structure and the functionality of the system. Thus, with this document functional system model is creating whose application specification is independent from implementation details. On the next development stage, the model evolves by adding to it features of a specific platform. Finally, in this phase the Java code generation for Android platform is automatically done from Unified Modeling Language diagrams. This application is running on Android operating system in the OMAP3530 processor.

Keywords: Android, Embedded System, Methodology, Model-Driven Development.

1 Introduction

The rise of complexity in embedded systems applications at the present time has been considerable. The new applications require accomplishing demands of quality factors such as: good performance, reliability, security, portability, interoperability, robustness, scalability and low power. Thus, these factors introduce an additional effort in order to fulfill the development of reliable systems, with development time and cost reasonable.

In order to face this problem, Model-Driven Development (MDD) approach is introduced as design software methodology. This approach focuses on the modeling of functionality of the system without consider the technology in which it will be implemented in order to manage the complexity.

Typically MDD methodology defines four models types during the development process: Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM) and Platform Specific Implementation (PSI). The first model defines the system requirements. The second one defines the functional system model. The third one defines the software model with details of the platform and the last model correspond to the application generated in a target language, that implement the functionality defined in the previous models [1].

On the other hand, Unified Modeling Language (UML) has been adopted as the de facto standard modeling language in the MDD approach. For Embedded System (ES) design, UML provides a rich set of constructs to support the modeling of system functionality, behavior and structure. Some limitations of language related to ES design such as the modeling of real-time requirement have been addressed with the use of UML profiles that permit extend the expressiveness of the such language [2].

In this work, we applied a software development methodology for embedded systems. We used a MDD approach to design a medical device for health monitoring in critical situations, called SIMMIT. The application is running on Android [3] operating system and the OMAP3530 platform.

This article is organized as follows. Chapter 2 presents some related work. Chapter 3 describes the OMAP3530 platform and tools, Chapter 4 provides an overview of the methodology used. Chapter 5 describes the case study. Finally the conclusions and future work are presented in Chapter 6.

2 RELATED WORK

In [4], proposed a methodology based on MDD approach to the development of mobile applications implemented in a cell phone. This work has developed a graphical modeling language specific to mobile applications, and coming up with a generic algorithm for the conversion of this graphical model into code. The main effort was putted in the design of interaction techniques, which will allow creating mobile applications easily. Unlike this work, authors presents a list of guidelines for modeling mobile applications, regardless platforms or the context.

Another work [5] proposes model-driven development of mobile personal healthcare applications. The authors
developed an approach in order to modeling care plans for chronic disease, using two domain-specific visualizing languages (DSVLs). The first allows healthcare providers to model complex care plans, health activities, performance measurements and sub-care plans. The second DSVL describes a mobile device interface for the care plan. A code generator synthesizes mobile device implementation of this care plan application. Unlike these works, in our application communicates with a medical devices which is managed from the communication and processing of physiological variables.

3 PLATFORM AND TOOLS

DESCRIPTION

Android mobile platform is supported by Texas Instrument’s OMAP3530 processor. The OMAP generation of high-performance, applications processors are based on the enhanced device architecture and are integrated on TI's advanced 45-nm process technology. This architecture is designed to provide best in class ARM and Graphics performance while delivering low power consumption. This balance of performance and power allow the device to support a huge variety of multimedia applications [6].

The OMAP3530 integrates a GPP (General Purpose Processor) ARM Cortex™-A8 @600MHz, a DSP (digital signal processor) TMS320C64x @430MHz plus a graphics accelerator 2D and 3D PowerVR SGX 530. The GPP controls all hardware resources using a generic operating system like Linux, Windows CE or, in this case, Android. The DSP acts as coprocessor of GPP. It also integrates various peripherals and interfaces to connect the different types of external devices [7].

The tools using are the following:

3.1 Eclipse

Eclipse is a software development kit (SDK) [8] consists of the Eclipse Platform, and Java development tools and a multi-language software environment composed by an integrated development environment (IDE) with an extensible plug-in system. Eclipse can be used to develop applications in various programming languages including Ada, C, C++, Java, Perl and Python. Development environments include the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++, among others.

3.2 Android Software Development Kit

The SDK (Software Development Kit) for Android is officially supported with Eclipse Plug-ADT (Android Development Tools plugin) and includes a set of development tools. This SDK including: debugger, libraries, phone virtual machine, documentation, sample code and tutorials [9].

3.3 IBM Rational Rhapsody

Rational Rhapsody is a modeling environment based on UML [10]. It was primarily designed to accelerate development and reduce costs, improve quality and managing complexity, through visualization of the models. Moreover IBM Rational Rhapsody helps to maintain consistency across the development life cycle to facilitate agility in response to requirements.

4 METHODOLOGY

The current trend is to use the methodologies in high levels of abstraction in the early stages of design, in order that the description of the system can be quickly and completely [11]. MDD (Model Driven Development) approaches have been proposed as a clear methodology for developing embedded systems [12], because provides the ability to streamline, standardize and replicate design practices by allowing a completely independent functional specification to implementation.

MDA (Model Driven Architecture) is one of the most popular MDD approaches proposed by OMG (Object Management Group) for software development, based on models at different levels of abstraction. In the MDA approach, a system is modeled using a platform independent model (PIM), which is transformed into a platform specific model (PSM) using design patterns. The languages used to express these models are defined by means of meta-models that are able to define abstract and concrete syntax and operational semantics of modeling languages.

The main objective of MDA is to separate the functionality of the architecture to build flexible systems, which do not depend on a hardware platform, or a specific software architecture, i.e. the application PSM can easily generated for different platforms, using the same PIM.

This work defines a process for generating a concurrent application Code which will run on the OMAP3530 hardware platform. From a PIM application, where define its functionality, is necessary generate a PSM model that considers the low level details necessary for the application to run properly on the target platform. Such details are closely tied to the hardware platform, in our case it is necessary to consider: Android operating system. The process consists of 4 stages:

A. Development of requirements: In this stage it is necessary to select the requirements that will guide the subsequent phases; i.e. the actors, their responsibilities, some functional features and constraints.
In the case study, a set of requirements is identified in the first iteration. Some of them were selected to establish the base architecture. These are related to the critical or main functions of the system. Although this application development could require more iteration, so the design is projected to be scalable.

**B. Application Model:** This stage consists in develop a PIM from the requirements document. In this phase is necessary identify basic functional features of the system, in that order the uses case and general classes are identifying from requirements document and generated using a modeling environment based on UML Rational Rhapsody.

**C. Application Specific Model:** This stage describes the functionality of the application, along with non-functional features of the system is obtained by enriching the PIM model by means of a transformation process. The transformation PIM to PSM is probably the most common focus in MDA. This is traditionally the mapping from the essential analysis model to the platform-specific design model. In this part the PIM model is transformed semi-automatically using meta-modeling transformation techniques to generate output for the design phase according to the platform to implement. Moreover, design patterns are applied manually in a PIM which transform in PSM.

**D. Code Generation:** With the PSM fully developed, you can feed a tool that transforms the PSM model described in UML to code c, c + + or java. The code is generated with Rational Rhapsody tool and subsequently the particular modifications are introduced in the code directly. It is important to emphasize that it is possible to accomplish the iterative development process due to the Rational Rhapsody tool capability for keeping the consistency between models and code.

**E. Running the application:** Once the code generation process is complete, proceed to test the application on either a virtual or on the OMAP3530 platform using Android operating system.

## 5 CASE STUDY

The proposed methodology is evaluated by developing a prototype called SIMMIT (System Integration Medical Monitoring and Interoperability for Telecare). The objective is to integrate and transmit towards a medical center the medical record information of a patient in emergency state within a medical assistance vehicle or in a remote station.

The signals derived from monitoring equipment (ECG, heart rate, respiratory rate, oxygen saturation and blood pressure) and recording the findings in the patient should be integrated and appropriate to the patient's electronic medical records in a standard format for then sent to a remote location via a communication network when the medical staff required. In Figure 1 illustrates the SIMMIT and its environment.

In order to implement the case study was carried out the methodology proposed in paragraph 4. First they drew up a requirements document, from there took place a model of the application. In Figure 2 illustrate a use-case diagram of SIMMIT.

With the use cases and after an object analysis performed a conceptual class diagram in Figure 3 illustrates a portion of the class diagram.
In order to verify the PIM, was made a diagram of state machine, as shown in Figure 4.

Once the PIM is checked, proceed to perform the PSM. This model must take into account the nature of the application described in the PIM, and the details to implement it on a specific platform. In our case we use the OMAP3530 platform and Android operating system. The PSM should include design patterns, as these provide unified solutions to recurring software problems. In Figure 5 illustrates how the class diagram has been modified in order to add the features of Android and the observer pattern [13].

The next phase is to generate code from the PSM with the help of the tools IBM Rational Rhapsody and Eclipse, as shown in figure 6. The final step is to run the application designed with the MDD methodology on the OMAP3530 platform as is showing in Figure 7. If everything is correct it has completed the design process, otherwise is necessary proceeds with an iterative process that consists of refining the model PSM and generates code again to fulfill all the system requirements.
6 Conclusions

In this work a software application for medical health care called SIMMIT has been developed using a MDD approach with success. This methodology starts with UML modeling application functionality regardless of implementation details, via an intermediate model that adds these details up to the executable code generation for OMAP3530 platform with Android 2.2 operative system.

The methodology aims to improve the productivity of embedded system design, rescuing the benefits of reusability, scalability, maintainability and modularity of system components and it provides the ability to separate the functionality of the application from the implementation details by means of models.

ACKNOWLEDGMENT

We would like to express our thanks to the Excellence research Center, ARTICA and to the members of Microelectronic research group and Software Engineering Research Group from Antioquia University.

7 References


