GUI code generation for Android applications using a MDA approach

Conference Paper · November 2012

DOI: 10.1109/ICoCS.2012.6458567

CITATIONS
18

READS
636

3 authors, including:

Ayoub Sabraoui
University Ibn Zohr - Agadir
10 PUBLICATIONS **46** CITATIONS

SEE PROFILE

Ismail Khriss
Université du Québec à Rimouski UQAR
29 PUBLICATIONS **295** CITATIONS

SEE PROFILE

GUI Code Generation for Android Applications Using a MDA Approach

Ayoub SABRAOUI, Mohammed EL KOUTBI
Mobile Intelligent Systems team (MIS)
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS)
Rabat, Morocco
a.sabraoui@uca.ma, elkoutbi@ensias.ma

Ismail KHRISS
Département de mathématiques, d'informatique et de génie
Université du Québec à Rimouski (UQAR)
Québec, Canada
Ismail.khriss@uqar.ca

Abstract—**Developing applications for mobile platforms is a compound task, due to variability of mobile OSs and the number of different devices that need to be supported. Model-Driven Architecture (MDA) approach could provide a possible solution to provide an automated way to generate a Graphical User Interface (GUI) for such applications. In this paper, we propose an approach based on MDA, to generate GUI for mobile applications on smartphones. The adopted approach consists of three main steps (i) analyzing and modeling the GUI under UML; (ii) transforming the obtained diagrams to a simplified XMI schema using JDOM API; and (iii) generating the GUI based on MDA. Our method has the advantages to generate automatically GUI for several platforms, and gives a graphical way for designing in UML.**

Keywords- UML; MDA; Metamodel; GUI for mobile applications; Mobile;

I. INTRODUCTION

Because of the large number and variety of available mobile operating systems and smart devices, developing the same application for these different platforms, becomes a tedious task. The variety in the features of devices, including: GUI, processing speed, and GPS sensor, is another constraint that must be taken into account when developing mobile applications. Model-Driven Architecture (MDA) approach [1] could provide an elegant solution to offer an automated way to generate some parts of an application targeting different mobile platforms.

The Object Management Group (OMG) adopted the MDA as an approach for model-driven engineering. In this approach, models have taken more focus in the development process. MDA separates the modeling task from the implementation details, without losing the integration of the model in a target platform. The key technologies of MDA are UML (Unified Modeling Language) [2], MOF (Meta-Object Facility) [3], (XMI) XML Meta-Data Interchange [4] and QVT (Query/View/Transformation) [5].

The MDA approach defines models at three different levels of abstraction: Computation Independent Models (CIM), Platform Independent Models (PIM), and Platform Specific Models (PSM).

According to the MDA standard, model transformation aims to provide a mean to specify the way to produce target models from a number of source models. Model transformations are composed of fine-grained rules that transform elements defined in a specific source metamodel into other elements of the target metamodel as illustrated in "Fig. 1".

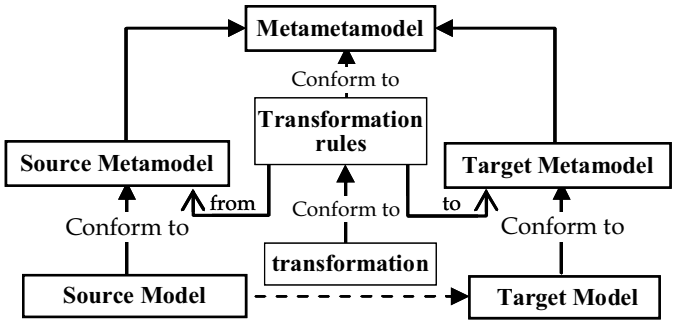


Figure 1. Transformation process in MDA approach

Model transformations are defined at the metamodel level. The set of rules and constraints (to obtain the target model from the source model) refer to elements of the metamodels. Consequently, transformations can be applied to any source model conforming to the source metamodel and produce a target model conforming to the target metamodel.

On the other hand, UML has been widely accepted as the standard modeling language that covers a large and diverse set of application domains. UML 2.2 has 14 types of diagrams divided into two categories. Seven diagrams are interested by the structural aspect of an application, while the remaining seven diagrams focus on its behavior aspect.

In this work, we propose a new approach to design the GUI for mobile applications by defining a mobile platform independent model, and we adopt the MDA approach to generate the GUI for a specific operating system. Basically, our approach consists of three main steps. The first one consists of analyzing and modeling the GUI using class and object diagrams based on the specified requirements. In the second step, these models are transformed to a simplified XMI files using the JDOM API. In the third step, the MDA approach is

978-1-4673-4766-2/12/$31.00 ©2012 IEEE

adopted to automatically transform the defined models, from the independent platform to a specific one. The MDA transformation rules are expressed in the Atlas Transformation Language (ATL) [6].

Compared to existing approaches, ours offers an automatic way to generate GUI for mobile applications for different mobile devices. In this paper, we focus more on Android OS platforms by defining its PSM. Another advantage of our approach is that it offers an easy and graphical way to design GUI in UML.

This paper is organized as follows. The second section presents some related works. The adopted GUI development approach is described in the third section. The fourth section shows the applicability of our approach through a case study. The last section concludes the paper and presents some future works.

II. RELATED WORK

The variety of exiting mobile platforms, diversity of devices and their multiple features; present a major challenge to develop mobile applications in such environments. In this section, we discuss some related works.

The approach presented by [7] is the closest to our work, in the sense that, the authors propose a method to develop the same functionalities, starting with the user interface, for different mobile platforms. They define metamodels for the target platforms and an abstraction model for the source platform, and the mapping is done by the graph transformation. However, they don't offer an automate process to generate code from models.

There is another stream of works, which propose metamodels for specific mobile platforms such as Android platform [8], and Windows Phone 7 [9]. These works enable designing platform specific models for mobile application. In fact, special characteristics of each platform are explicitly presented when modeling a mobile application. In contrast, our approach permits platform independent modeling and automatic GUI code generation for different mobile platforms.

In [10], authors develop Domain-specific Languages (DSL) for modeling mobile device applications and propose a graphical model transformations rules to map between Domain-specific models (DSMs) and the generated applications. They propose a structured approach for code generation where layered model transformations are used to modularly isolate, compile and re-combine various aspects of DSMs. This approach still requires some improvements to automatically generate transformation rules.

III. DESIGNING GRAPHICAL USER INTERFACE FOR MOBILE APPLICATIONS

Applications have structural and dynamic descriptions. To cover the structural aspect, we propose UML diagrams (class diagrams and object diagrams) to describe and design GUI for mobile applications. Our approach proposes a metamodel to design a platform independent model of the GUI of a mobile application. Then, the MDA approach is applied to generate the GUI code targeting a specific mobile platform. "Fig. 2" presents the various steps characterizing our method.

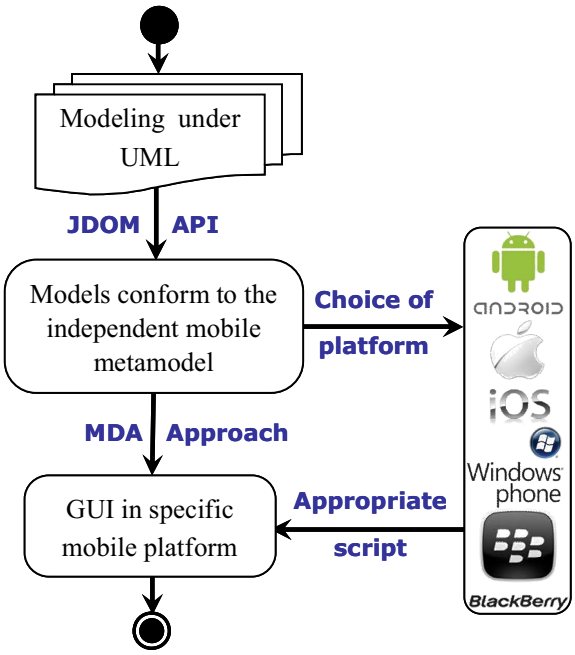


Figure 2. Our method to design GUI for mobile applications

A. *Analysing and modeling under UML diagrams*

In this step, object diagrams of the GUI of a mobile application are built from its specified requirements according to the proposed metamodel as illustrated in "Fig. 3". In order to cover the diversity of mobile OSs, this metamodel is conceived to be independent of specific mobile platforms.

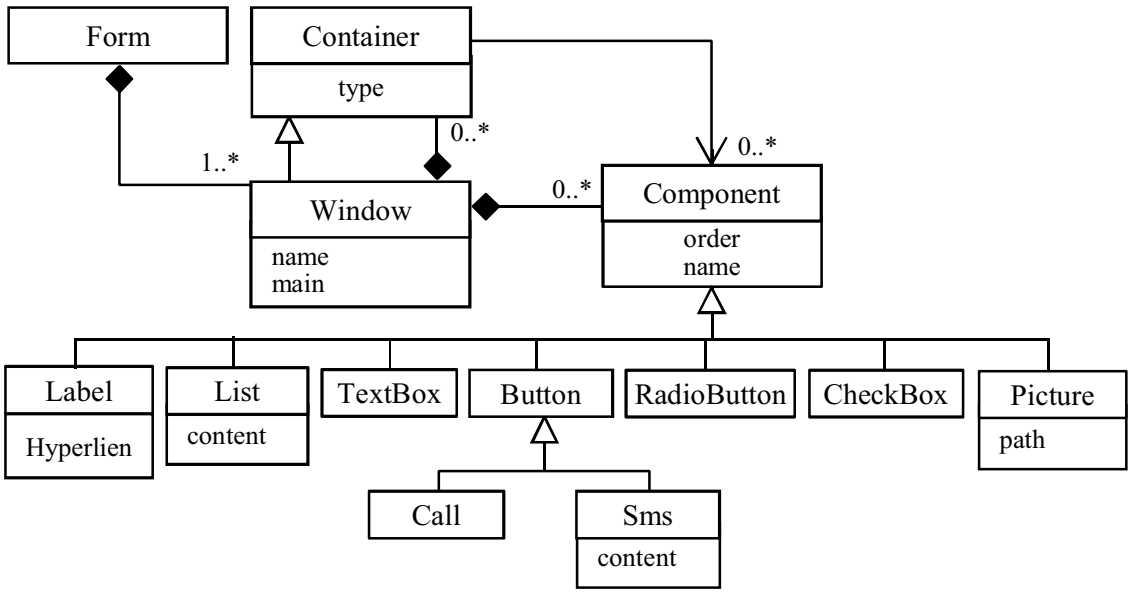


Figure 3. A mobile application metamodel

B. *Transforming the obtained document to a simplified XMI schema*

The resulting object diagram obtained in the previous step, models the GUI for mobile application independently of the mobile platform. This diagram is in XMI format, and needs a transformation to eliminate unnecessary complexity from its representation. This first mapping is done using the JDOM API.

C. *Applying MDA approach to generate GUI in a specific platform*

In this step, we adopt the MDA approach to transform platform independent model of the GUI model to a specific platform (Android, iOS, Windows Phone, or BlackBerry OS…) as shown in "Fig. 4". In this paper, we discuss the GUI generation for Android platforms. This transformation requires the metamodels of the two models: source (independent mobile platform) and target (specific one). The transformation rules are expressed using the ATL language.

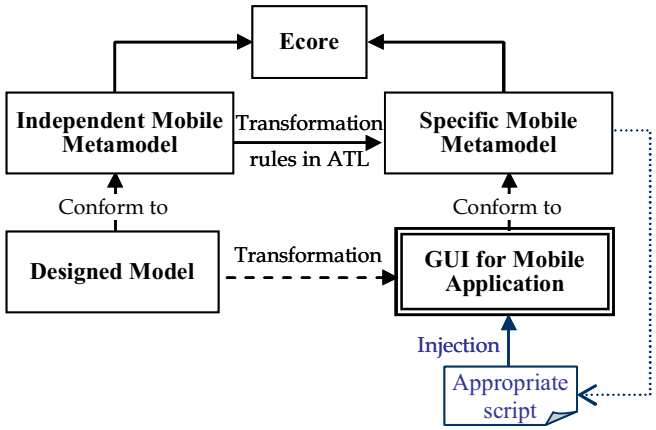


Figure 4. MDA approach to generate GUI in a specific platform

The choice of the target platform is followed by an injection of a corresponding script, which contains some standard attributes and their default values.

IV. Case study: A. Dr. Phil's Personality Test

In this section, we present an example that allows us to illustrate our approach that automatically generates GUI for Android OS. We consider a psychological test named "A. Dr. Phil's Personality Test" available on the website [11]; which let a user know how others perceive him. In this test, the user should honestly answer the questions about himself. These questions are structured in different windows (GUI). The application automatically scores the quiz, and gives an immediate result to the user.

A. *Analysis and modeling phase*

The main goal in this step is to design the different GUI for our mobile application using the UML object diagrams. These diagrams are in accordance with the mobile platform independent metamodel as illustrated in "Fig. 3".

In the "Dr. Phil's Personality Test" application, we define about 12 windows. The first is the main window, the last correspond to the test results, and intermediate windows present different questions and possible answers that the user can choose.

The designer should model these requirements using an object diagram respecting the metamodel in "Fig. 3". In this example, the window consists of containers and components, and containers contain themselves other components.

"Fig. 5" shows an extract of the object diagram, corresponding to "Dr. Phil's Personality Test" application.

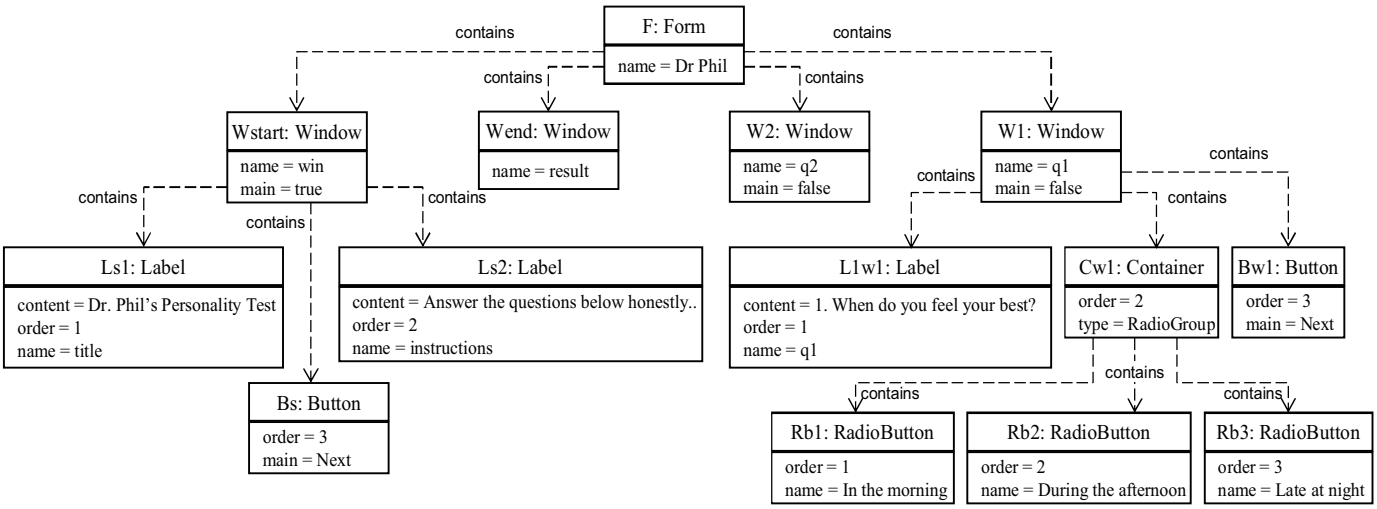


Figure 5. The object diagram extract of Dr. Phil's Personality Test

The resulting object diagram is built under EclipseUML tool, and it is in XMI format. An important step in our method is to transform this diagram to a simplified XMI file in order to simplify the next steps of our method.

B. MDA based development approach to generate GUI for Android OS

In this step, we adopt the MDA approach to transform the previous object diagram representing the platform independent GUI model of our example into a model specific to the Android platform. For this mapping, we introduced an Android GUI metamodel as shown in "Fig. 6", and we defined the corresponding rules between the two metamodels using ATL language.

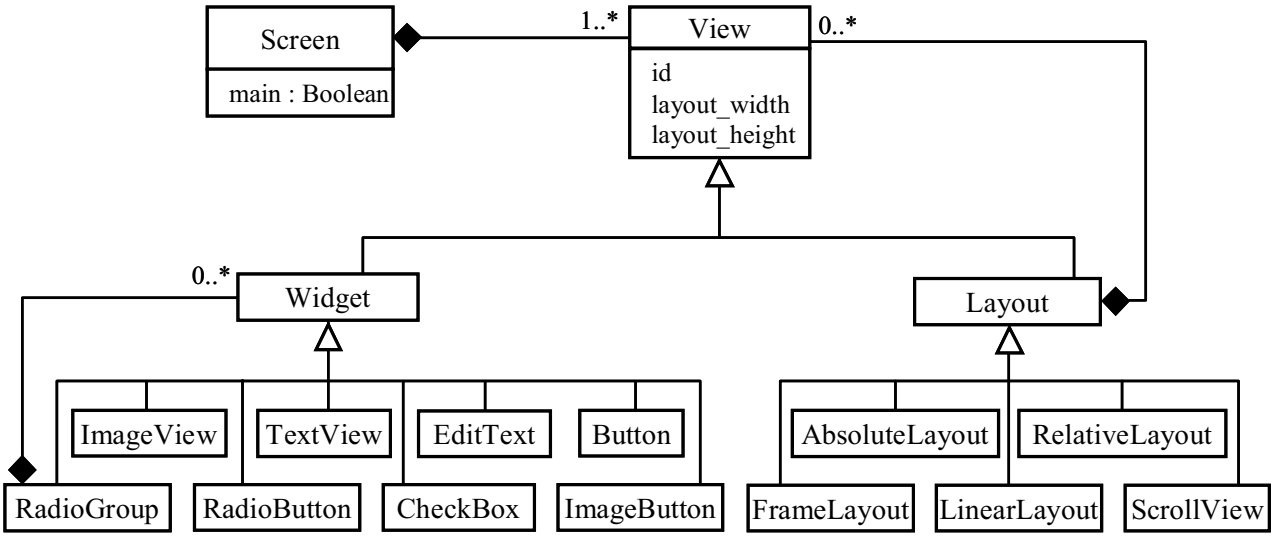
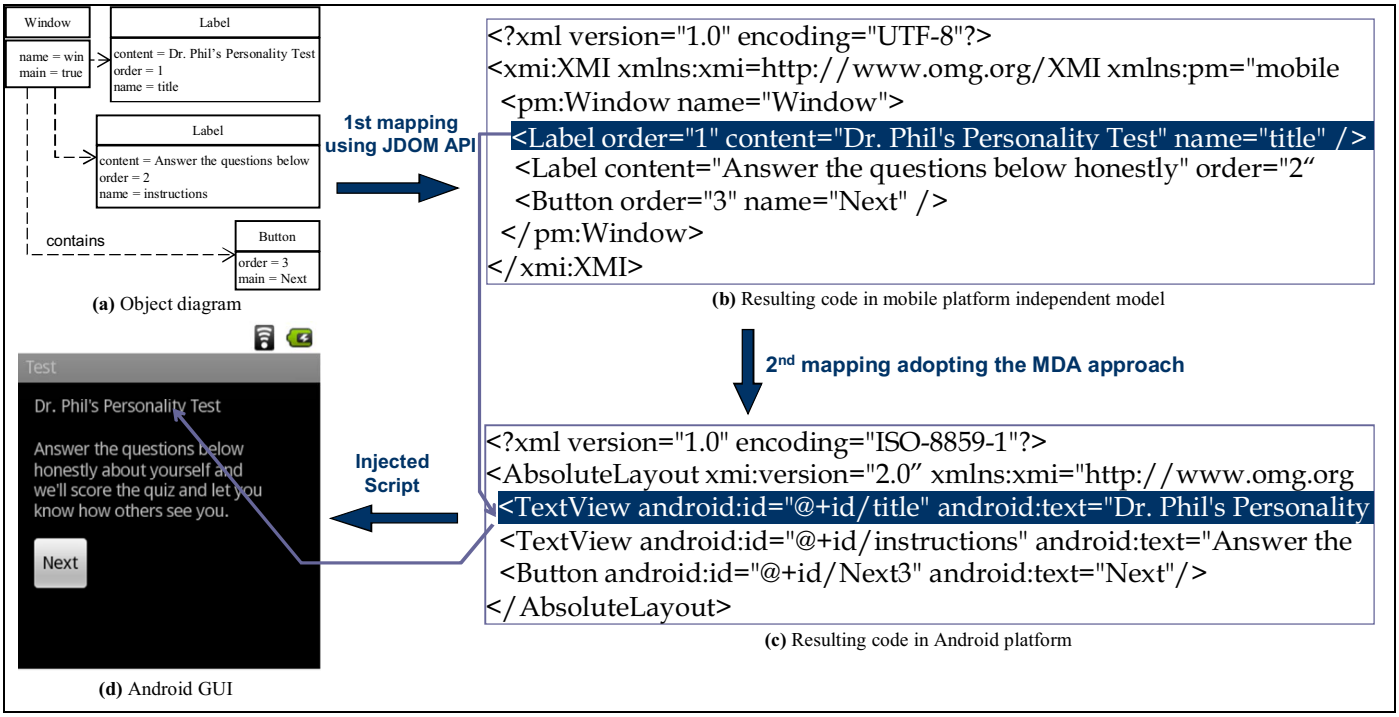


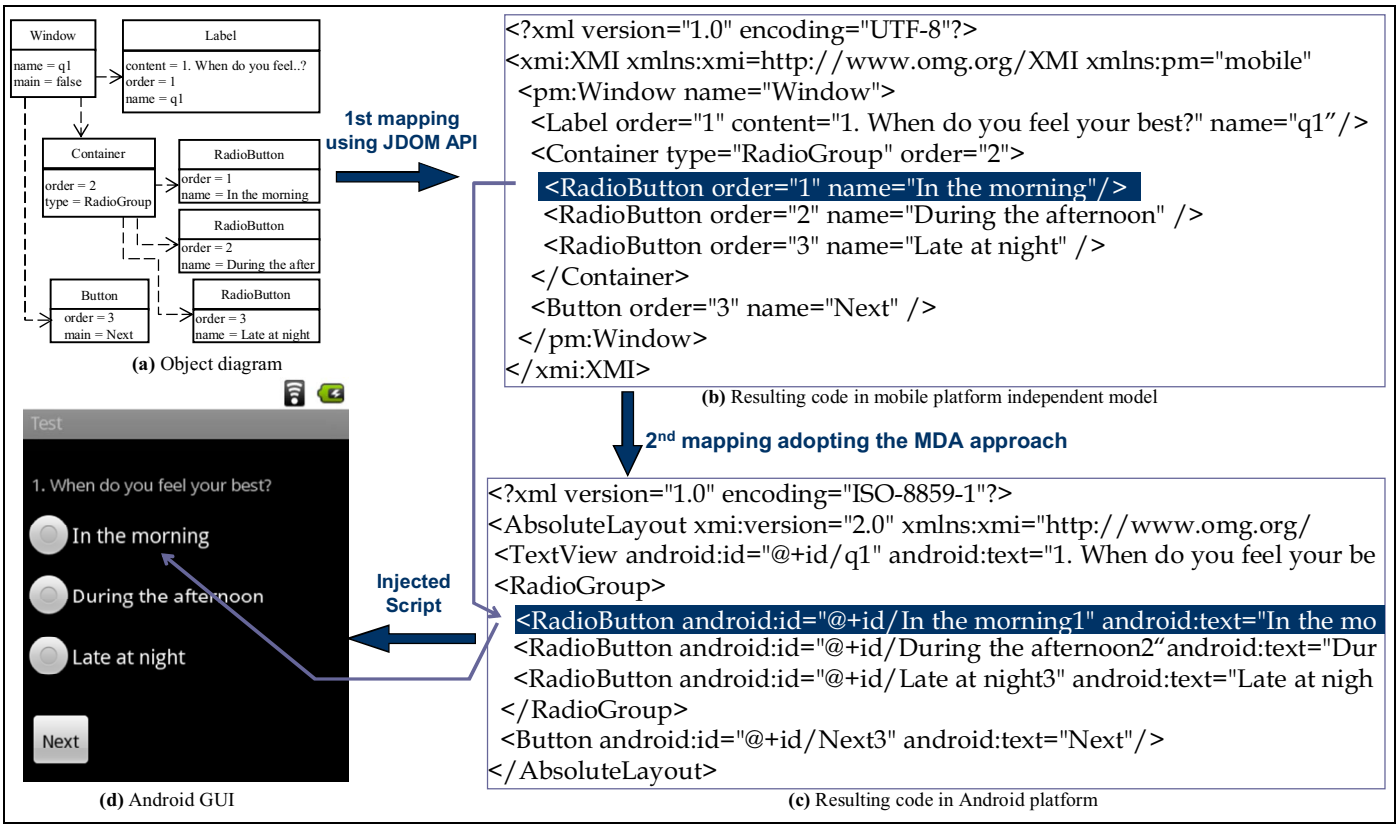
Figure 6. An extract of the Android application metamodel

C. GUI code generated targetting the Android platform

We depict the different steps of the process of GUI generation for "A. Dr. Phil's Personality Test" as an Android application. "Fig. 7" illustrates the different steps of our approach.



The main screen



Another screen

Figure 7. The different steps of our method to GUI code generation for an Android platform

The different steps of the process of GUI generation for an Android application can be summarized as following:

(a): The UML object diagram which depict the designed model.

(a)→ (b): Corresponds to the 1st mapping from the object diagram to a simplified XMI.

(b): The resulting model which is conforms to the mobile independent platform.

(b)→ (c): Corresponds to the 2nd mapping from the mobile platform independent mobile to the mobile platform specific one. In this case, the Android OS.

(c): The generated GUI code is compatible to the Android platform.

(c)→ (d): Personalizing the generated GUI code by adding some attributes in order to enhance the usability of the GUI.

(d): The execution of the generated personalized GUI code for the Android platform.

V. CONCLUSION AND FUTURE WORK

This paper introduced a new MDA based development approach to generate the Graphical User Interface (GUI code) for mobile applications from UML diagrams. For this, we first proposed a mobile platform independent model to design GUI independently of the mobile target platforms (Android, iOS, Windows phone 7…) and we adopted the MDA approach to automatically generate the GUI for a specific platform, by defining its corresponding metamodel. This separation of levels of abstraction can construct a model to automatically generate GUI to various mobile systems.

Modeling under UML concepts and notation is a great advantage of our approach, by offering a graphical way to design mobile applications, and making easy the communication between designers, developers and others.

For future improvements, we will study the possibility to extend our method to take into account the behavioral aspects of mobile applications.

We will also work on supporting all operating systems by analyzing and defining the features of each platform.

REFERENCES

[1] Object Management Group, "MDA Guide Version 1.0.1", omg/2003-06-01, June 2003.

[2] Object Management Group, "OMG Unified Modeling Language (OMG UML) Superstructure, Version 2.2" http://www.omg.org/spec/UML/2.2/Superstructure/PDF, 2009.

[3] Object Management Group, "OMG, Meta Object Facility (MOF) specification, Version 1.4", http://www.omg.org/, 2002.

[4] Object Management Group, "OMG MOF 2 XMI Mapping Specification, Version 2.4.1", http://www.omg.org/spec/XMI/2.4.1, 2011.

[5] Object Management Group, "Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification", Version 1.1, January 2011, http://www.omg.org/spec/QVT/1.1/

[6] ATLAS group LINA & INRIA, "ATL: Atlas Transformation Language", Version 0.2", http://www.uio.no/studier/emner/matnat/ifi/INF5120/v05/undervisningsmateriale/, 2005.

[7] I. Madari, L. Lengyel, and T. Levendovszky, "Modeling the User Interface of Mobile Devices with DSLs", 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, November 15-17 2007, pp. 583-589

[8] K. Minhyuk, S. Yong-Jin, M. Bup-Ki, K. Seunghak, and K. Hyeon Soo, "Extending UML Meta-model for Android Application", 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, May 30 2012-June 1 2012, Shanghai, China, pp.669-674

[9] B. Min, M. Ko, Y. Seo, S. Kuk, and H. S. Kim, "A UML metamodel for smart device application modeling based on Windows Phone 7 platform", TENCON 2011 - 2011 IEEE Region 10 Conference, 21-24 November 2011, Bali, pp. 201-205

[10] R. Mannadiar, and H. Vangheluwe, "Modular synthesis of mobile device applications from domain-specific models", Proceedings of the 7th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, New York, USA, 2010, pp. 21-28

[11] J. M. Grohol, "Dr. Phil's Personality Test", Mar 2010, http://psychcentral.com/personquiz.htm