

Volume 40, No. 1
January 2015
ISSN 1548-7717
CODEN AACL

Advanced and Applied Convergence Letters

AACL 04

Advanced and Applied Convergence

**1st International Joint Conference, IJCC 2015
Ho Chi Minh City, Vietnam, February 2015
Revised Selected Papers**



International Institute for Business and Convergence



International Partnership for Advanced and Applied Technology

Table of Contents

Improving Utilization of GPS Data for Urban Traffic Applications / 1

Nguyen Duc Hai, Nguyen Tan Phuc, Doan Khue, Ta Ho Thai Hai, Pham Tran Vu, Huynh Nam and Le Thanh Van

Ontology-based Context Modeling for Smart Home Domain / 5

M. Robiul Hoque, M. Humayun Kabir, Toshiro Minami, Sung-Hyun Yang

Modeling of a Context-Aware System for Smart Space / 8

M. Humayun Kabir, M. Robiul Hoque, Sung-Hyun Yang

Optimal Placement of Medical Robotic System Using Genetic Algorithms / 11

Quoc Cuong Nguyen, Youngjun Kim, HyukDong Kwon

An Automatic Mechanism of Ui Code generation for iPhone Platform / 15

Hyun Seung Son, Woo Yeol Kim, R. Young Chul Kim

Stepped Impedance Resonator Filter for Cognitive Radio System / 19

Seong Ro Lee

Feedback Linearization for Non-linear Time Varying System / 21

Jong-Yong Lee, Kye-dong Jung, Seongssoo Cho

Development of Vuforia VR Platform Based miniature Geobukseon / 23

Chul-Seung Yang, Jeong-gi Lee, Han Byul Kang, Dae-Won Park, Beomjin Kim, Sang-Hyun Lee

An Approach for Scheduling Problem in Port Container Terminals:Moving and Stacking / 26

HA Phuoc Lan, LE Ba Toan, HUYNH Tuong Nguyen , NGUYEN An Khuong, NGUYEN Van Minh Man

Apparatus for displaying search results for keyword inputted by user with multi formatted data indexingmethodinbigdataplatfrom / 31

Wooyung Lee, Daesu Chung, Jeong-Jin Kang, Young-Dae Lee, Joon Lee

A Study of Software Defect Rate Estimation Technique in Complete Repeat Testing Environments / 34

Young B. Park, Mahmoud Tarokh, R. Young Chul Kim

A Framework of the 3D Geofence System for Location Awareness / 37

Byungkook Jeon, DORJ Ulzii Orshikh, Sungjin Cho, Sungkuk Cho

A Study on LED Emotion Light Control Method Using Moving Mean Filter / 40

Soonho Jung, Junwoo Kim, Minwoo Lee, Seungyoung Yang, Jaekwon Shin, Jintae Kim, Kyoungwha Yoon, Juphil Cho, Nguyen Quoc Cuong, Jaesang Cha

An Automatic Mechanism of UI Code generation for iPhone Platform

Hyun Seung Son*, Woo Yeol Kim**, and R. Young Chul Kim*†

* SE Lab, Dept. of CIC(Computer and Information Communication), Hongik University,
Sejong Campus, 339-701, Korea

e-mail : {son, bob†}@selab.hongik.ac.kr

** Department of Computer Education, Daegu National University of Education,
Daegu, 705-715, Korea

e-mail : john@dnue.ac.kr

Abstract

In this time, it is difficult to develop software for the heterogeneous smartphone platforms (Android, iPhone, MS phone) at a time due to the difference between Application Programming Interface (API) in each different platform. It is a complicated question how to reduce cost of software development without reduplication. In approaching the issue, one of solutions will be the automatic code generation for the each smartphone platform. It can reduce the unnecessary time of code on software development, and also improve the quality of code through reusing the verified codes. In this paper, we extend our previous research that automatically generates the User Interface (UI) code for Android platform. So, we propose the extended method for iPhone platform. The extended method consists of 1) the transformation step of transforming class diagram from the abstracted UI model, and 2) the generation step of generating UI code from the class diagram. Our approach is possible to rapidly develop the smartphone application by reducing the amount of programming code.

Keywords: Automatic code generation, User Interface (UI), Reusability, iPhone, Objective C, Model Driven Development (MDD)

1. Introduction

In software development process, the best solution for rapid development is reusability of software. To realize the reusability consists of two ways: 1) using a software platform, and 2) automatic tools. First, using a software platform is possible in only one environment because of dedicated hardware device. Therefore, this solution doesn't development for heterogeneous software. Second, automatic tools are necessary to be able to perform in your developing environment. If the automatic tools exist for your development environment, you are able to develop software rapidly. In otherwise case, you will be unhappy because it takes a long time to make the automatic tools. Specially, this situation happens when developing more heterogeneous platforms than a single platform.

To develop the software development in a smartphone uses a platform due to these reasons. It is good method in a single platform, but don't use other platform in heterogeneous device. So, it is difficult to develop the heterogeneous smartphone platforms (Android, iPhone, MS phone) at a time for the difference that software of each platform use dissimilar Application Programming Interface (API). To solve this problem, we select one of solutions that be the automatic code generation using tools for the each smartphone platform. Automatically generating code gives us to reduce time of software development because it automatically generates code and reduces unnecessary coding time for developers. Also, it can improve the good quality of the software through re-using the verified codes. But we need new approach

against taking a long time to make the automatic tools.

Model Driven Development (MDD) [1] automatically transfers *platform specific model* (PSM) from *platform independent model* (PIM). To transfer to PSM(s) from PIM, the model transformation in MDD is a core technique, which consists of two ways: 1) Model-to-Model, and 2) Model-to-Text. We will focus on Model-to-Text transformation that is a method to automatically form a code from the model. This method algorithm is classified with two type mechanisms: visitor-based and template-based [2]. The visitor based mechanism is composed of purifying the internal expression of model tree and writing text in text stream such as Jamda [3]. Jamda provides the assembly of classes that represent UML model. Template-based method is more similar with the code than the visitor mechanism using OMG's Model to Text Transformation Language [4]. This is easily used in the repeated development of template such as Acceleo [5]. Acceleo can represent text work due to the approach presented in this section as well as code pieces within text that are incorrect in syntax or meaning.

In order to generate the full code with the result of our previous research [6-14], three elements of model transformation was required such as the UI to configure the mobile screen, the code that performs functions, and the project to configure the development environment. In this paper, we are limited to propose the automatic generation method of UI code in iPhone platform to improve and extend a part of our previous research [15]. The proposed method is consisted of two steps: 1) transfer class diagram from the abstracted UI model and 2) generate UI code from the class diagram. If the method is applied, it is possible to rapidly develop the smartphone application by reducing the amount of code that the programmer must write.

This paper is organized as follows. Chapter 2 mentions the method of UI code generation for iPhone platform. Chapter 3 mentions the conclusion and future work.

2. Automatic Mechanism of UI Code Generation for iPhone Platform

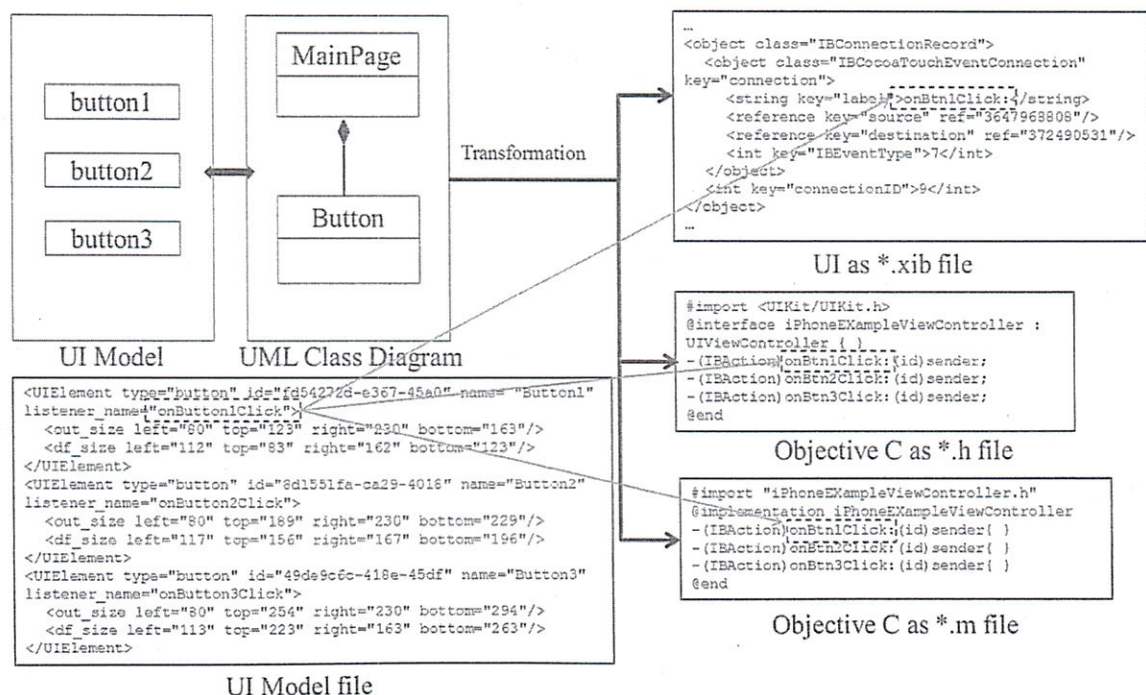


Figure 1. A method of code generation for User Interface (UI) in iPhone platform

If you are develop the application that perform some commands when you click the button in smartphone platform, it is necessary to write the code showing button on screen and to write event handler function to perform commands. In case of Android platform, you have to write the code to register event handler

function. But, UI designer registers the event handler function on UI file (such as xib) in a case of iPhone platform. Therefore, the iPhone platform has to save the event handler function on UI file without writing the code to register event handler function. For this reason, the code generation in Android platform makes only java files, but the iPhone platform generates the UI and code at the same time. Also, the code of iPhone platform must have the header and source files. Therefore, transforming of Model-to-Text is required three files such as UI file, header file, and source file.

In the view of this limited situation, we propose the method as shown in figure 1. The proposed method is consisted of two steps: 1) transfer class diagram from the abstracted UI model and 2) generate UI code from the class diagram. In the first step, we design UI model for developing the application using the abstracted UI model. Our automatic tool generates the XML Metadata Interchange (XMI) file with the designed UI model, and transfers to the class diagram from the UI file. In UI Model, specifying the name of the event handler adds into the function of class in class diagram. In the second step, model transformation generates the files such as UI (.xib), header (.h), and source (.m) at the same time from the information of UI Model as shown on the arrow of figure 1. The name of the event handler effects the three files such as Xib file (generated by UI designer), h file (that is, header), and m file (that is, source in the code of Objective-C).

3. Conclusions

The automatic code generation gives to shorten development time due to reduce unnecessary coding time of his/her developer, which can improve the quality of the software with re-using the verified codes. The existing code generation just generates the skeleton code of UML class diagram included just structural information, but needs additional written code. In this paper, we are limited to propose the automatic generation method of UI code in iPhone platform to improve and extend a part of our research. The proposed method consists of two steps: 1) to transfer class diagram from abstracted UI model and 2) to generate UI code from class diagram. This method increase code generate rate with class diagram, UI model, and event handler profile. Further research should be conducted, which is not dealt in this study on model and UI transformation for heterogeneous platform.

Acknowledgments. This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS (Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"] and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2011601).

References

- [1] B. Selic, "The pragmatics of model-driven development," *Software, IEEE*, Vol. 20, Issue 5, pp. 19-25, 2003.
- [2] K. Czarnecki, S. Helsen, "Feature-Based Survey of Model Transformation Approaches," *IBM Systems Journal*, Vol. 45, No. 3, pp. 621-64, 2006.
- [3] Jamda, The Java Model Driven Architecture 0.2, <http://sourceforge.net/projects/jamda/>
- [4] OMG, MOF Model to Text Transformation Language, v1.0, formal/2008-01-16.
- [5] Obeo, Acceleo User Guide, <http://www.acceleo.org/>
- [6] W.Y. Kim, H.S. Son, J.S. Kim, R.Y.C. Kim, "Development of Windows Mobile Applications using Model Transformation techniques," *Journal of KIISE : Computing Practices and Letters*, Vol. 16, No. 11, pp. 1091-1095, 2010.
- [7] W.Y. Kim, H.S. Son, J.B. Yoo, Y.B. Park, R.Y.C. Kim, "A Study on Target Model Generation for Smartphone Applications using Model Transformation Technique," *International Conference on Internet (ICONI) 2010*, Vol. 2, pp. 557-558, 2010.
- [8] W.Y. Kim, H.S. Son, J.S. Kim, R.Y.C. Kim, "Adapting Model Transformation Approach for Android Smartphone Application," *Advanced Communication and Networking*, Springer CCIS 199, pp. 421-429, 2011.
- [9] W.Y. Kim, H.S. Son, R.Y.C. Kim, "A Study on UML Model convergence Using Model Transformation Technique for Heterogeneous Smartphone Application," *Software Engineering, Business Continuity, and Education*, CCIS 257, pp. 292-297, 2011.

- [10] W.Y. Kim, H.S. Son, R.Y.C. Kim, "Rule Extraction Method for Model Transformations in Heterogeneous Smartphone Applications," *Information Journal*, Vol.16, No.1(B), pp. 615-626, 2013.
- [11] W.Y. Kim, H.S. Son, R.Y.C. Kim, "Design of Code Template for Automatic Code Generation of Heterogeneous Smartphone Application," *Advanced Communication and Networking*, Springer CCIS 199, pp. 292-297, 2011.
- [12] H.S. Son, W.Y. Kim, R.Y.C. Kim, "Concretization of the Structural and Behavioral Models based on model Transformation Paradigm for Heterogeneous Mobile Software," *International Journal of Software Engineering and Its Applications*, Vol. 7, No. 4, pp. 389-399, 2013.
- [13] H.S. Son, J.S. Kim, R.Y.C. Kim, "SMTL Oriented Model Transformation Mechanism for Heterogeneous Smart Mobile Models," *International Journal of Software Engineering and Its Applications*, Vol.7, No.3, pp. 323-331, 2013.
- [14] H.S. Son, W.Y. Kim, R.Y.C. Kim, "A Design of Metamodel for Model Transformation of Heterogeneous Smartphone UI," *Proceedings of the 16th Korea Conference on Software Engineering (KCSE 2014)*, Vol. 16, No. 1, pp. 173-180, 2014.
- [15] H.S. Son, W.Y. Kim, R.Y.C. Kim, "An Automatic UI Code generation for Android Mobile Platform," *Proceedings of Advanced and Applied Convergence & Advanced Culture Technology*, AACL 03, pp. 151-154, 2014.