ISSN 2287-3252

ICCT 2015

"The 5th International Conference on Convergence Technology 2015"

Vol.5 No.1

- **Date : June 29 – July 2, 2015**
- **Place : Chateraise Gateaux Kingdom Sapporo Hotel, Hokkaido, Japan**
- **Co-organized by :**
  - Korea Convergence Society
  - Korea Institute of Science and Technology Information
  - The Korean Association for Comparative Government
  - The Society of Digital Policy & Management
  - Convergence Society for SMB
  - Konyang Univ. Well-Dying LAB
  - Korea Mobile Enterprise Promotion Association
  - DAEHAN Society of Industrial Management
- **Sponsored by :**

KISTI NTIS Division & GSDC Center, LG Hitachi Co., Ltd, ALLforLAND Co., Ltd, KYUNGBONG Co., Ltd, SOFTITECH Co., Ltd, TAEJIN Infortech Co., Ltd, Geo Matics Co., Ltd, MIJU C&D Co., Ltd, R2soft Co., Ltd, Hanbit Academy, Inc., Korea IT Consulting Co., Ltd, Open Link System Co., Ltd, SungWon-IT Co., Ltd, SJ info&communications Co., Ltd, Neighbor system Co., Ltd, Able IT Co., Ltd, INFORMADE Co., Ltd, SelimTSG Co. Ltd, KORNEC Co., Ltd, MTData Co., Ltd, Mobile Law Co., Ltd, DELTASYSTEM Co., Ltd, Daewoo Information Systems Co., Ltd, LG CNS Co., Ltd, ICTWAY Co., Ltd, GFT Co., Ltd, QbizOn Co., Ltd, NANUS Information Co., Ltd, Hankyung I-NET Co., Ltd, VITZROSYS Co., Ltd, Duplex Co., Ltd, Maverick Systems Co., Ltd, Bizmerce Co., Ltd, DAWON ICT Co., Ltd, INNOZIUM Co., Ltd, MetaBiz Co., Ltd, Human Information Co., Ltd, AtechIns Co., Ltd, Comtec System Co., Ltd

KCS KOREA CONVERGENCE SOCIETY
http://www.kcons.or.kr

Session 5-D (SW Visualization (SE Center)) Chair Seung Yeob Yu(Namseoul Univ.)

● 14:30~15:50 Tuesday June 30, 2015

01. W-07-06_Computer Simulation on HPDC Process by Filling and Solidification Analysis / 360
Tae-Hoon Yoon(Namseoul Univ., Korea), Hong-Kyu Kwon(Namseoul Univ., Korea)

02. W-13-09_Extracting Software Architecture based on Reverse Engineering / 362
Woo Sung Jang(Hongik Univ., Korea), Chae Yun SEO(Hongik Univ., Korea), R. Young Chul Kim(Hongik Univ., Korea), Woo Yeol Kim(Daegu National Univ. of Education, Korea), Young Soo Kim(NIPA, Korea)

03 W-13-10_Internal Code Visualization for Analyzing Code Complexity / 364
So Young Moon(Hongik Univ., Korea), Sang Eun Lee(NIPA, Korea), R. Youngchul Kim(Hongik Univ., Korea)

04. W-13-11_Replacing Source Navigator with Abstract Syntax Tree Metamodel (ASTM) on the open source oriented tool chains for SW Visualization / 366
Hyun Seung Son(Hongik Univ., Korea), So Young Moon(Hongik Univ., Korea), R. Young Chul Kim(Hongik Univ., Korea), Sang Eun Lee(NIPA, Korea)

05. W-13-12_Requirement Tracking Visualization for Validating Requirement Satisfaction / 368
Bokyung Park(Hongik Univ., Korea), Haeun Kwon(Hongik Univ., Korea), Young Soo Kim(NIPA, Korea), R. Young Chul Kim(Hongik Univ., Korea)

06. W-13-13_Mobile Based Testing with Code Visualization / 370
Keunsang Yi(Hongik Univ., Korea), Hyeoseok Yang(Hongik Univ., Korea), R. Young Chul Kim(Hongik Univ., Korea)

07. W-33-06_Content Analysis of Green Advertisements in Korea / 372
Mi-Jeong Kim(Hanyang Univ., Korea), Sangpil Han(Hanyang Univ., Korea)

08. W-33-09_Online Public Opinion Dissonance between Korean and Chinese Netizens: its Causes, Functions and Solutions / 374
JiHye Lee(Namseoul Univ., Korea), SeungYeobYu(Namseoul Univ., Korea)

The 5th International Conference on Convergence Technology 2015
June 29 - July 2, 2015, Chateraise Gateaux Kingdom Sapporo Hotel, Hokkaido, Japan

Mobile Based Testing with Code Visualization

[1]Keunsang Yi, [2]Hyeoseok Yang, [*3]R. Young Chul Kim
[1,2,*3 Corresponding Author] *SELab., Dept. of Computer and Information Communication, Hongik University, Sejong, Korea, {Keunsang, Yang[2,] bob[*3]}@selab.hongik.ac.kr*

Abstract Android platform applications are developed with two separated ways: UI design and source code. It's very hard to analyze mobile SW complexity by module, class, and package units which are the existing measuring elements of SW complexity. To solve this problem, we suggest to visualize *an activity call graph* between activities which are the basic unit of android platform, and measure the complexity between intents for analysis of code complexity of mobile software. By code visualization through implementing A Tool-Chain, we can build up *an activity call graph*, and to analyze malfunction of android mobile codes.

Keywords: *Code Visualization, Mobile Based Testing, Complexity, SW Quality, SW Visualization in Nipa's Software engineering Center*

1. Introduction

In these days, mobile SW is required to be developed, and works on various mobile platforms with original functions which are developed in the existing development environment. These mobile SW need highly advanced and efficient resources. Especially poor optimization of advanced smart devices makes it difficult to develop mobile SW. Code complexity becomes increased and it leads to an increase in SW defects of devices.

Android mobile applications are developed with UI code and the core code apart. For a static analysis, A Tool-Chain visualizes source code based on the existing SW visualization [1]. It's impossible to visualize mobile SW code expressing interactions between activities by intent. We reinforce visualization of internal structure of mobile application SW through intent between android activities. With this visualized structure, we diagnose the reasons of Bad-smell structure for refactoring, and finally suggest an automatic static analysis of mobile application SW through reassignment of modules.

Therefore, it is organized as follows: Chapter 2 describes mobile application testing technique as related research, Chapter 3 shows automatic static analysis mechanism, and Chapter 4 mentions to utilize Tool-Chain for a static analysis of mobile application code. Chapter 5 describes conclusion and future research.

2. Related Research

Since mobile applications work in various platform environments, mobile application testing is different from the existing SW testing. To meet requirements of mobile application testing, it carries out testing activities and objectives like functional and behavior testing, interoperability testing, usability testing, and so on. In order to test mobile SW accurately, each application on various platform devices would run. However, testing all the devices would take great expense. Mobile application testing includes Emulation-based Testing, Device-based Testing, Cloud Testing and Crowd-based Testing.

3. Automatic Static Analysis Mechanism

On development of android mobile applications, it can be described with mutual relationship of class and activity.

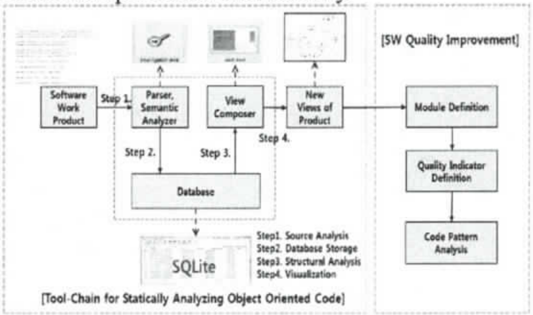

Figure 1. Improving SW Quality through SW Tool-Chain

It is important to work activities in the android mobile devices. Data transferring and calling between activities is delivered using the object called intent. The existing visualization Tools, which can't analyze intent-based activity calls, are not suitable in the android-based mobile development environment. Figure 1 shows Improving SW Quality through SW Tool-Chain.

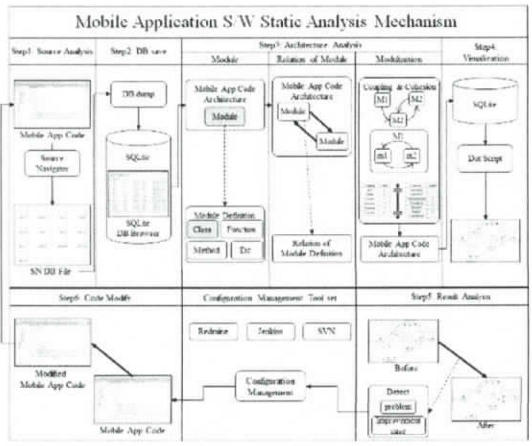

Figure 2. Process Mechanism for Automatic Static Analysis of Mobile Applications

4. A Tool-Chain for Analyzing the Mobile Software Code

We use each open source tool comprising Tool-Chain in this approach [1,2] such as Parser (Source Navigator 6.0(SN)), Database(SQLite), and View Composer(DOT). Use Case Diagram Drawing Tool is based on JAVA.

When analyzing mobile source codes through parsing of source navigator, SN DB FILE is extracted. By DBdump this extracted SN DB FILE is transformed into text and saved in SQLite database. Each SN DB FILE has peculiar information to fit its role. Each information is saved in tables and analysis of each column is carried out. To do Query Call for Extraction of Coupling between Activities, we analyze codes through the source navigator which is extracted as SN DB files. After this information of SN DB files is saved in SQLite database. Each SN DB file saved in database has information of mobile application source codes to match its role. Its needed information is extracted through SQLite query based on coupling example codes. Figure 2 mentions process mechanism for Automatic Static Analysis of Mobile Applications.

Activity Call Visualization: unlike the existing Java environment, one of characteristics of the android environment is communicating with users through activities. Activities are the basic units constituting user interface, and according to users' demand, work activities which are sequentially called. They are called through intent, and it's hard to analyze them through the existing analyzing method of Java code.

5. Conclusion and Future Research

This study suggests building up automatic static analysis mechanism for mobile application SW testing, and methods to reduce defects of mobile application SW through activity call graph visualization.

A static analysis oriented Tool-Chain process is suggested, and it can enhance quality control and visualization by the existing static analysis of SW code. Android codes are divided into internal function and UI. When the existing Tool-Chain process analyzes android mobile application SW, it is important to call and transfer data between activities as the basic unit of android UI that can't be analyzed and visualized. We suggest setting up mechanism for basic static analysis of SW, and an activity call graph of android mobile application SW. This measure reduced coupling of mobile application SW, and stopped unnecessary calls and data transfer by visualizing calls between activities. For future study, we suggest activity-related UI design, and reduce cyclomatic complexity of coupling & cohesion.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2011601) and Research and Development Service through the Telecommunications Technology Association (TTA) funded by the National IT Industry Promotion Agency (NIPA).

References

[1] Bokyung Park, Haeun Kwon, Hyeoseok Yang, Soyoung Moon, Young Soo Kim, R. Young Chul Kim, "A Study on Tool-Chain for statically analyzing Object Oriented Code", KCC2014, KIISE, pp.463-465, 2014.

[2] www.nipa.kr

[3] Haeun Kwon, Bokyung Park, Keunsang Yi, Young B. Park, Young Soo Kim, R. Young Chul Kim, "Applying Reverse Engineering through extracting Models from Code Visualization", The 2014 Fall Conference of the KIPS, KIPS, USA, vol. 21, no. 2, pp.650-653, 2014