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)

Requirement Tracking Visualization for Validating Requirement Satisfaction

[1]Bokyung Park, [2]Haeun Kwon, [3]Young Soo Kim, [*4]R. Young Chul Kim
[1,2,*4] *SELab., Dept. of Computer and Information Communication, Hongik University, Sejong, Korea, {park[1], kwon[2,] bob[*4]}@selab.hongik.ac.kr*
[3] *NIPA, Seoul, Korea, Country, ysgold@nipa.kr*

Abstract Our previous research proposed the inner structured visualization in the object-oriented code, and the improvement process of software qualities[1]. But this method was difficult to trace and validate requirements. To solve this problem, this paper refines the extension of the previous visualization approaches, that is, *Requirement Tracking Model* to validate requirements which can ensure how many requirements are achieved during/after developing/developed a system.

Keywords: *Requirement Rate, Requirement Tracking, NIPA's SW Visualization*

1. Introduction

Our domestic software in the developing environment has some difficulties to control software qualities due to invisibility and to increase complexity. They also have focused on code-centered approaches for a rapid development. In order to improve software quality, it are needed development management and maintenance, and reverse engineering techniques, and can also identify the correlation between modules inside the code to show code complexities. The previous study proposed the visualization of the internal structure of the object-oriented code, and the process of software quality improvement [1]. In this way, it is possible to identify the inner structure of the existing code with the visualized mechanism, and also to improve software qualities by defining the module measurement with 'coupling & cohesion' metrics, and the quality indicators, and performing refactoring against the original source code. Before this approach, it cannot trace and validate requirements. So this paper suggests the requirement tracking process for validating the requirement satisfaction.

2. Software Visualization Process

Fig. 1 is software visualization process for a quality improvement [1,2]. This process (Tool-Chain Mechanism) is implemented to combine with each open source oriented tool such as Parser, Database, Open Source, and View Composer. In addition, it is composed of four steps which have included source code analysis, DB storage, structural analysis, and visualization to perform with independent functions [3]. The quality improvement measures software qualities based on coupling & cohesion.

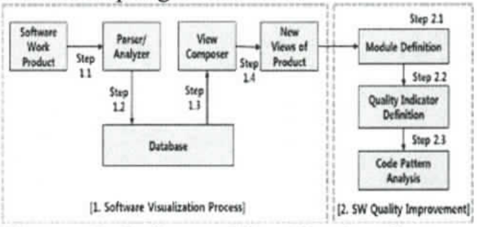

Figure 1. Software Visualization Process

3. Requirement Tracking for Validating Requirement Satisfaction

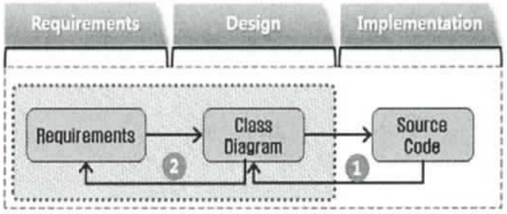

Figure 2. Requirement Tracking Visualization for Validating Requirement Satisfaction

The process consists of extracting designs from source code Class, and also identifying requirements via Design from code. For example, extracting Class Diagram through tracing the inner structure based on requirement traceability in a code. Fig. 2 describes Requirement Tracking Visualization for Validating Requirement Satisfaction. In Fig. 2, ① is a UML Class Diagram extraction process [4]. This step extracts the coupling & dependencies between classes, and visualizes a class diagram by performing the process of Fig. 1, ② defines the mapping relationship between the extracted Class Diagram and requirements, and specifies it on the coupling graph. These requirements compare and analyze the requirement list which is defined in 'Redmine' of the Tool-Chain mechanism and Java Source Code, and is specified with Requirement Number on the coupling graph. Then Requirement Rate is extracted to validate the mapping relation between a class diagram and requirement in order to validate it. The degree of requirement satisfaction is the value obtained by dividing the number of the detected requirements with the total number of the whole requirements.

$$\text{Requirement Satisfaction Rate} = \frac{\textit{The Number of Detected Requirements}}{\textit{The Number of Total Requirements}} \quad (1)$$

4. Case Study

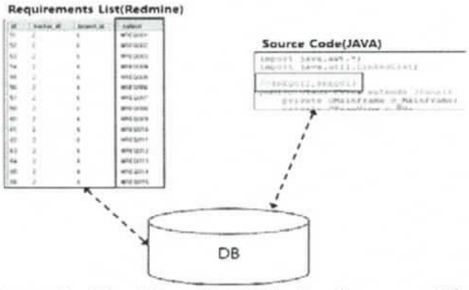

Figure 3. The Mapping Relation between Class diagram and Requirements

The Case study applied the code of Use case drawing tool Himem which was previously developed. Based on analyzing the mapping relation, we defines the requirements described in Redmine of Tool-Chain and Java Code. Java Code which corresponds to the requirements, is analyzed, and is commented on the part where the appropriate code begins(// @ Requirement number). When the Tool-Chain is executed, the requirements of Redmine and the part specified in the source code are compared to the information stored in the database. If the appropriate information is existing in DB, it is specified in the class diagram with this information, but if not, it is not done. In Fig. 4, the requirement numbers are represented as Class(CBasicFunctionBar<-> REQ006/007, CMainFrame<->REQ001), and the extracted information is used to calculate the degree of requirement satisfaction. Then the total number of requirements is 29, and the number of mapped requirements is 15. Therefore, the requirement satisfaction degree is 52% in the applied case. This result can show that only such percent reflects the Application Case Tool.

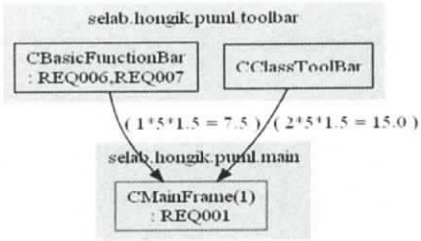

Figure 4. Mapping Relations

5. Conclusion

In order to develop a high-quality software, a systematic method is needed for requirement validation, and this paper is suggested a requirement tracking process for its verification. This process visualizes requirements associated with Design, that is, a class diagram. Also, it is possible to ensure how many requirements are achieved by extracting requirement traceability in the system development. The future study will be about a variety of quality measurement methods including 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, Hyun Seung Son, Young Soo Kim, Sang-Eun Lee, R. Young Chul Kim, "A Case Study on Improving SW Quality through Software Visualization", Journal of KIISE, KIISE, vol. 41, no. 11, pp.935-942, 2014.

[2] www.nipa.kr

[3] 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.

[4] 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.