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, Atechlns 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)

Internal Code Visualization for Analyzing Code Complexity

[1]So Young Moon, [2]Sang Eun Lee, *[3]R. Youngchul Kim,
*[1,*3]Dept. Computer and Information Communication, Hongik University,*
Sejong Campus, Korea, {msy, bob}@selab.hongik.ac.kr*
[2] NIPA, Korea, selee@nipa.kr

Abstract The challenging issues of software quality remains rarely addressed, e. g., invisibility, increasing complexity and unfavorable development environment in small businesses, which impedes proper software quality management. Mostly, existing legacy systems fail to preserve their original design, while increasing their code complexity due to more patching of the original codes. To solve such problems, we adopt a code visualization technique which substantially reduces the code complexity between modules. For this, we suggest a tool chaining method based on the existing open source software tools, which extends NIPA's Software Visualization techniques.

Keywords: *Reverse Engineering, Tool-Chain, SW Visualization*

1. Introduction

Software has been widely used across diverse fields, serving as a key to add values to final products to ensure their competitive edge. In contrast to the increasing importance of software, its invisibility and complexity as well as domestic SME (Small and medium enterprise) s' software development environment have thwarted software quality management [1].

This paper intends to contribute to high-quality software development by focusing on visualization of core domains of software, namely, visibility of development process, reduction of complexity, and the absence of documentation about development and design. In addition, source codes underpinning software need be updated in time to reflect up-to-date information for the operability of software, whilst the quality must be kept at the highest level, as software can be explained by its source codes only [2]. Thus, this paper applies the Software Visualization Technique developed by the NIPA(National IT Industry Promotion Agency) with a view to: 1) detecting, altering and modifying the problems of legacy codes; 2) providing a guideline for rectifying software developers' bad habits by applying a reverse engineering technique via code visualization; and 3) coping with the absence of developers or relevant documentation to help maintain legacy systems. To visualize the internal structure of codes, this paper constructs a tool-chain by connecting a range of open sources.

This paper mentions the following chapters. Chapter 2 describes software visualization and reverse engineering with related studies. Chapter 3 presents complexity of inner structure. Finally, chapter 4 describes a discussion and a future work.

2. Software Visualization

NIPA's software visualization may be fit for high-quality software development of IT venture startups, SMEs and even established entities constrained by a lack of personnel and financial resources [3]. SW visualization aims to manage source codes and development processes, specifically involving visualization and documentation as a means of managing the quality of SW development. An entire process of software development needs to be efficiently managed to produce valuable software. It takes clear-cut goal setting, efficient fulfillment, on-going monitoring and control activities to successfully manage software development.

3. Complexity of Inner Structure

1) Definition of Module

The module definition step defines a module unit suitable for the target software code to be visualized. This paper defines classes as modules.

2) Definition of Quality Indicator

In designing software, inter-module coupling needs minimizing whereas inter-module cohesion needs in-creasing to develop high-quality software. Thus, quantitative measurement indicators for coupling and cohesion need to be set [4]. Here, coupling refers to inter-dependence or inter-relation between two modules. High inter-module coupling means

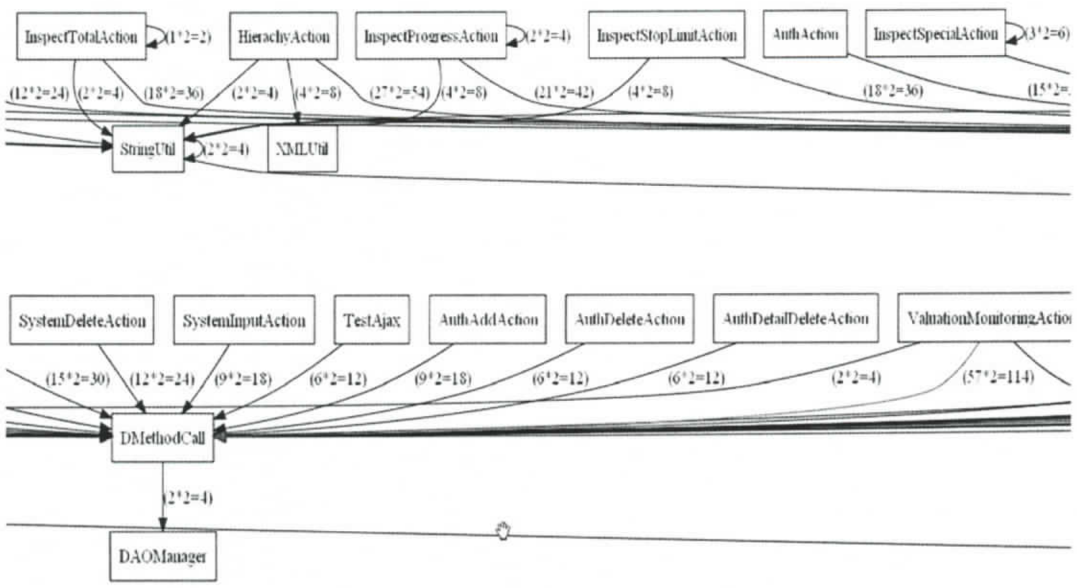

Figure 1. Internal Code Visualization from Analyzing Code Complexity

strong inter-dependence between modules, which has adverse effects on transformation, maintenance and reuse of modules. Independent modules require low inter-module coupling and dependence. Coupling is sub-divided into data, stamp, control, external, common, and content couplings. Inter-module dependence increases in the direction of the content coupling, while decreasing in the direction of the data coupling.

Figure 1 shows internal visualization from analyzing code complexity. Coupling of between Valuation MonitoringActions and DMethodCall is 144. This is a high coupling, but others are not too high. We use this internal visualization method, and we see the internal structure from source code.

4. Conclusion

For the purpose of developing high-quality software, this paper focuses on detecting and altering problems of existing codes, and rectifying software developers' bad habits. Because a means of delivering high-quality software even with highlighting development/ testing and maturity measurement leads developers to additional workloads other than development, it cannot be an alternative for enhancing the quality of legacy systems. To address this issue, the proposed tool-chain method defines modules, quantifies the complexity of codes based on software structures & the frequency of inter-module relations, and shows the quality of codes with quality indicators for inter-module coupling as part of software visualization. The proposed method enables even developers of bad habits to lessen the code complexity with refactoring.

Future research will deal with the visualization of software quality in terms of cohesion & coupling.

Acknowledgements. 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. This work was supported by the National Research Foundation of KOREA (NRF) and Center for Women In Science, Engineering and Technology (WISET).

References

[1] NIPA SW Engineering Center, "SW Development Quality Management Manual (SW Visualization)", 2013, pp. 3-4.

[2] Steve McConnell, Code Complete (2nd ed.), Microsoft Press, 2001.

[3] Geon-Hee Kang, R. Youngchul Kim, Geun Sang Yi, Young Soo Kim, Yong B. Park, Hyun Seung Son, "A practical Study on Code Static Analysis through Open Source based Tool Chains," KIISE Transactions on Computing Practices, vol. 21, no. 2, pp. 148-153, February 2015.

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