

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, MTDData 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

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)

Extracting Software Architecture based on Reverse Engineering

¹Woo Sung Jang, ²Chae Yun SEO, ^{*3}R. Young Chul Kim, ⁴Woo Yeol Kim, ⁵Young Soo Kim

^{1,2,*3} *SELab., Dept. of Computer and Information Communication, Hongik University, Sejong, Korea, {jang¹, bob^{*2}}@selab.hongik.ac.kr,*

⁴*Dept. of Computer Education, Daegu National University of Education john@dnue.ac.kr*

⁵*NIPA, Seoul, Korea, ysgold@nipa.kr*

Abstract It is very difficult to maintain the completed software without design and document. To easily maintain the code, we suggest to visualize the code with Open source oriented tool chain. As a result of this mechanism, we can easily reduce the maintenance costs, and create the clustering diagrams applied with source code with this tool chain mechanism based on reverse engineering approach. It is possible to improve the maintainability by reducing the coupling score & complexity. However, the more the software complexity increases, the more the clustering diagram complicates. So this diagram analysis is very difficult. This paper shows how to extract software architecture with analyzing the cluster coupling diagram based on the visualized tool chain. If applied software architecture in this tool chain, we may reduce analyzing time & costs, and show the whole code structure to increase readability. We implement the system using HTML and Java script.

Keywords: *Maintainability, Coupling, Software Architecture, Clustering Diagram*

1. Introduction

Recently, one property of software maintenance has become an important issue for quality maintenance. But the complete maintenance of software will spend high cost. If applied software reverse engineering with the software source code, we may reduce maintenance costs. If create the clustering diagram based on source code with reverse engineering, can be easily understand a source code structure. However, complex software has a complex inner structure. If the diagram structure may be complicated with class connection, it is difficult to maintain the software system. This paper applies to select the size level of cluster (such as component, model, and package) to show software architecture

within the whole code with the clustering diagram tool chain.

In this way, developer can recognize connections of classes, component, model, package, and subsystem. The diagram does not show to connect all classes, but a user can just select packages to show the cluster diagram with the coupling score between classes. As a result, class connected structure can be simplified. The User can improve the maintainability by reducing the clustering & coupling score between classes.

Chapter 2 describes the clustering oriented tool chain. Chapter 3 shows how to connect the selective software architecture. Chapter 4 mentions Results. Chapter 5 describes discussion.

2. The Clustering Oriented Tool Chain

The Clustering & Coupling Tool Chain was implemented in Jenkins[3][4]. When A User uploads source code, it creates the coupling diagram applied on Reverse Engineering. The diagram is configured only classes limited in this paper. The coupling score will mark between classes. Diagram Implementation shoes thus result in figure 1.

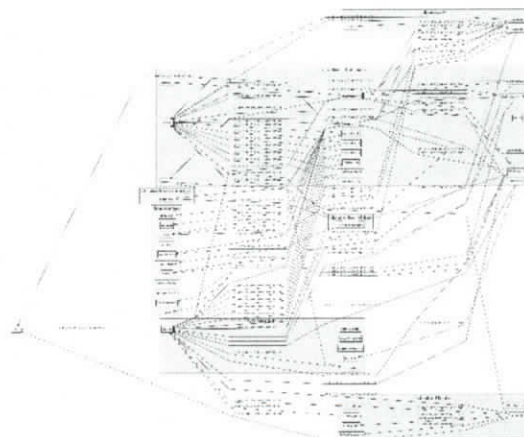


Figure 1. The clustering coupled diagram

3. Selective clustering diagram for Software Architecture

We add arguments for input function in the clustering oriented tool chain. Arguments type are like Table 1. Jenkins is a Web-based solution. Therefore, this system is implemented HTML and Java Script.

Table 1. Argument Type

Name	Function
Image File Directory	Graph Image File Location
Graph Size	Drawn Graph Size
Search Directory Location	Source Code Location
File Explorer	File List Viewer
BAT File Location	Tool Chain Location

4. Results

Users can enter the arguments in the environment based HTML, and see a file list in the file explorer. System Implementation shoes thus result in figure 2.

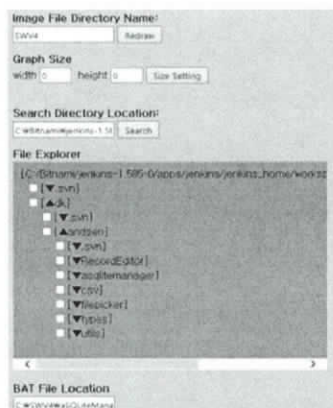


Figure 2. Arguments and file list

This approach can create each graph with classes (or package) selected by user. The graph generation is like this result in figure 3.

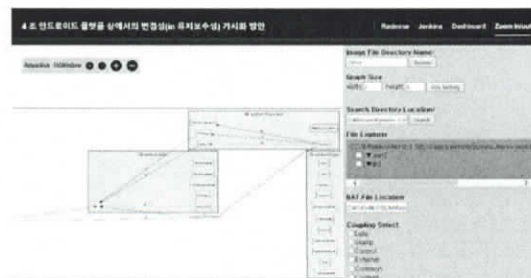


Figure 3. Reduced Diagram

5. Discussion

It is very difficult to maintain the completed software without design and document. To easily maintain the code, we suggest to visualize the code with Open source oriented tool chain. As a result, we can easily reduce the maintenance costs, and create the clustering mechanism applied with source code with this tool chain mechanism based on reverse engineering approach. It is possible to improve the maintainability by reducing the coupling score & complexity. However, the more the software complexity increases, the more the clustering diagram complicates. So this diagram analysis is very difficult. This paper shows how to extract software architecture with analyzing the cluster coupling diagram based on the visualized tool chain. If applied software architecture in this tool chain, we may reduce analyzing time & costs, and show the whole code structure to increase readability. The clustering oriented tool chain is draw with connection of all classes (or package).

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] Don Coleman, Dan Ash, Bruce Lowther, Paul Oman, "Using Metrics to Evaluate Software System Maintainability", IEEE, pp.44-49, 1994.
- [2] Mehwish Riaz, Emilia Mendes, Ewan Tempero, "A Systematic Review of Software Maintainability Prediction and Metrics", Journal Title, Third ESEM, pp.367-377, 2009.
- [3] <https://jenkins-ci.org>
- [4] <http://nipa.kr>