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

Session 5–C (SW Visualization (SE Center))   Chair R. Young Chul Kim(Hongik Univ.)

● 13:00~14:20   Tuesday June 30, 2015

01. W-13-01_Hybrid Visual Security Analysis to Prove Vulnerability for Safety / 262
    Seok Mo Kim(Dankook Univ., Korea), Sang Eun Lee(Software Engineering Center, Korea),
    Su Nam Jeon(Software Engineering Center, Korea), Young B. Park(Dankook Univ., Korea)

02. W-13-02_Compatibility Enhancing Agent for Managing Software Toolchain / 264
    Eun Seung Lee(Dankook Univ., Korea), Young Soo Kim(Software Engineering Center, Korea), Byungho Park(Hongik
    Univ., Korea), Young B. Park(Dankook Univ., Korea)

03. W-13-03_A Visualized Blocking Methtod againist a Hidden Malware in the Image / 266
    Byungho Park(Ministry of National Defence, Korea), R.Young Chul Kim(Hongik Univ., Korea), Young B. Park(Dankook
    Univ., Korea), Daecheol Shin(Korea e-Government Exp. Associations, Korea), Young Soo Kim(NIPA, Korea),
    SangEun Lee(NIPA, Korea)

04. W-13-04_A Guideline for Realization on extracting automatic size maturity level of diverse component
    via Source Codes / 268
    JunSun Hwang(Hongik Univ., Korea), R. Youngchul Kim(Hongik Univ., Korea), SangEun Lee(NIPA, Korea)

05. W-13-05_Design of a Flamework of 3D Geofence and Geocode / 270
    Jun Cho(Gangneung-Wonju National Univ., Korea), Kihyun Kim(Gangneung-Wonju National Univ., Korea),
    Jinhyung Park(Gangneung-Wonju National Univ., Korea), Sungjin Cho(Gangneung-Wonju National Univ., Korea),
    Byungkook Jeon(Gangneung-Wonju National Univ., Korea), Sungkuk Cho(Gangneung-Wonju National Univ., Korea)

06. W-13-06_Design of a Temporal Geofence System / 272
    Kihyun Kim(Gangneung-Wonju National Univ., Korea), Jun Cho(Gangneung-Wonju National Univ., Korea),
    Jinhyung Park(Gangneung-Wonju National Univ., Korea), Sungjin Cho(Gangneung-Wonju National Univ., Korea),
    Byungkook Jeon(Gangneung-Wonju National Univ., Korea), Sungkuk Cho(Gangneung-Wonju National Univ., Korea)

07. W-13-07_Extracting Designs via Code on Reverse Engineering / 274
    Haeun Kwon(Hongik Univ., Korea), Bokyung Park(Hongik Univ., Korea), R. Youngchul Kim(Hongik Univ., Korea),
    SangEun Lee(NIPA, Korea)

08. W-13-08_Extracting performance factors against performance degradation through Code Visualization
    / 276
    Geon-Hee Kang(Hongik Univ., Korea), R.Young Chul Kim(Hongik Univ., Korea), SangEun Lee(NIPA, Korea),
    Su Nam Jeon(NIPA, Korea)

Extracting Designs via Code on Reverse Engineering

[1]Haeun Kwon, [2]Bokyung Park, *[3]R. Youngchul Kim, [4]SangEun Lee

Dept. of Computer and Information Communication, Hongik University
Sejong Campus, 339-701, Korea,
{kwon, park, *bob}@selab.hongik.ac.kr
[4]National IT Industry Promotion Agency, Korea, selee@nipa.kr

Abstract The previous study suggested a too-chain mechanism for code visualization through reverse engineering techniques [2]. This paper proposes to extract design (object diagrams and sequence diagrams) based on a code analysis through such mechanism. That is, it intends to present coupling between class diagrams, its management method, and a mechanism for extracting sequence diagrams, and also track the generation process of coupling through it. It will be a foundation for a code, its design and requirement traceability.

Keywords: *Nipa's Software Visualization, Design Extraction*

1. Introduction

Due to the accelerated convergence, today's software has had their added functions and widened ranges so that they have increased the size of accidental damages caused by software defects [1]. In order to solve this problem, the previous research [2] extracted a class diagram based on Software Visualization Technique, and presented a management method if coupling between classes.

This paper is organized as follows: Chapter 2 introduces a extended method for coupling management studied previously, Chapter 3 explains how to extract a sequence diagrams in the object-oriented paradigm, and Chapter 4 , Conclusion shows how to apply the proposed process to the actual code, and also mentions future study.

2. Related work

The previous coupling management method research [2] is based on a software visualization technique. It was also a method to measure coupling from an object-oriented code and represented it as a class diagram, one of design documents. In addition, the coupling can be managed in the class diagram in correspondence of the dependency relationship between classes, and the coupling.
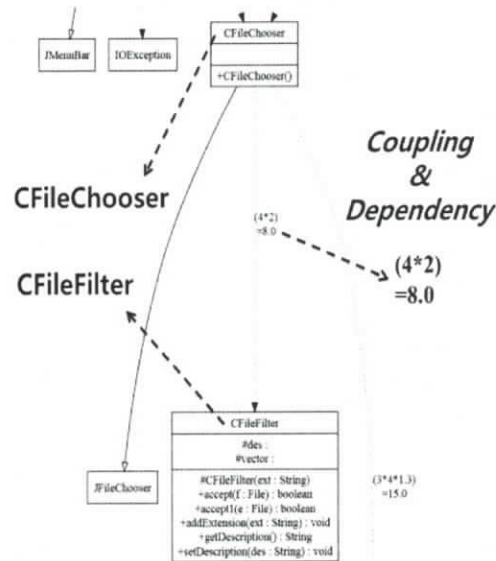

Figure 1. A coupling-applied class diagram

Figure 1 shows a coupling-applied class diagram generated by such process. It shows that the dependency relationship between CFileChooser Class and CFileFilter Class is the coupling value of 8.

3. Methods

The process to extract a sequence diagram from the implemented object-oriented code has four steps: 1) Source Analysis, 2) DB Storage, 3) Structural Analysis, and 4) Visualization like the previous study [2]. At first, Source Analysis is a step to decompose the given code into several components including classes, methods, and member variables. Decomposed components are sorted by factors in the next step of DB storage. Then the 3 step of Structural Analysis resolves calling sequences and coupling between factors.

In the 4 step of Visualization, objects are arranged in the sequence diagram in the order of object generation and placed messages in accordance with the sequence of method invocation calls. At this time, coupling is shown

in relation to the corresponded message, and then a seqeuntial diagram is generated with the application of coupling. Figure 2 shows a generation process of sequence diagrams with the application of coupling.
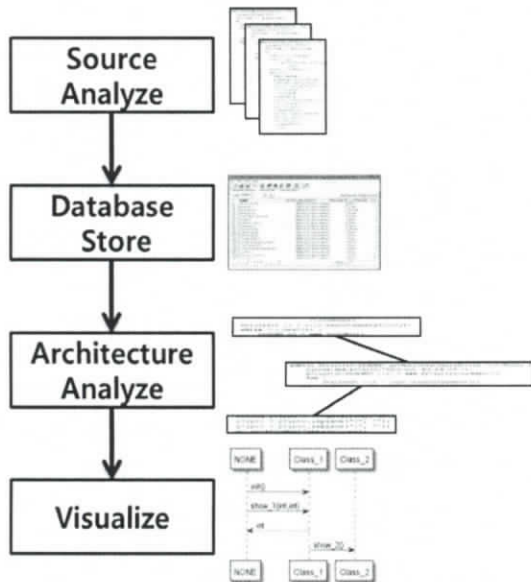

Figure 2. A generation process of sequence diagrams with the application of coupling

4. Result of Sequence Diagram

The code of "User case diagram drawing tool" was developed in Java language and applied to the proposed process, and its result is shown in Figure 3. It is a part of the whole sequence diagram and a visualization of the menubar package, and the formula above the message represents coupling. The coupling between CFileChooser class and CFileFilter class has the value of 8, and it is confirmed that it is corresponded to that in Figure 1. Also, actor is not an actual class but randomly made to indicate the starting point. Figure 3 shows a coupling-applied sequence diagram.
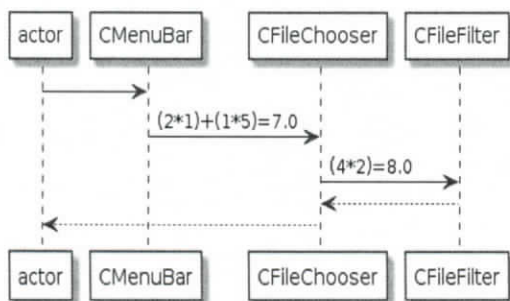

Figure 3. A coupling-applied sequence diagram

5. Discussion

This paper mentions to visibly identify the process where the coupling occurs in the extension of the previous study [2], and it applied the extraction and coupling of the object-oriented diagrams(sequence and object diagram) from a code. It visualized reverse engineering techniques, and proposed a design extraction from a code through it. Therefore, the future study can research a requirement traceability model and its automation, and also compensate for the insufficient information to extract a use-case based on such information.

References

[1] NIPA Software Engineering Center, "2013 Software Engineering White Paper", Korea, 2013

[2] Haeun Kwon, Bokyung Park, Keunsang Yi, Young B. Park, Youngsoo Kim, R. Youngchul Kim, "Applying Reverse Engineering through extracting Models from Code Visualization", KIPS, vol. 21, no. 2, pp.650-653, 2014