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-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 against 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)

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

Extracting performance factors against performance degradation through Code Visualization

[1] Geon-Hee Kang, *[2] R.Young Chul Kim, [3] SangEun Lee, [4] Su Nam Jeon
1, First Author, *2,Corresponding Author Dept. of CIC(Computer Information Communication), Hongik University, Sejong, Korea, {kang, bob}@selab.hongik.ac.kr
[3,4] NIPA, Seoul, Korea, {selee, snjeon}@nipa.kr

Abstract Nowadays, the SW industry in Korea considers the quality of SW in the viewpoint of its performance. It should also identify the various properties of its performance quality in accordance with the SW requirements. This paper proposes an identification method about the performance degradation factors through Code Visualization, and suggests a mechanism to visualize the extract factors. SW developers can improve the quality of SW performance and prevent coding practices to degrade SW performance.

Keywords: *Code Visualization, Nipa's SW Visualization, Performance,*

1. Introduction

Today's software industry has grown in and brought about the issue about its high quality. However, compared to the growing industry, a shorter period in market shipping has made the code centralized development for faster improvement in the scene of industry. As a result, software of poor quality have been mass-produced because the SW invisibility makes it difficult to manage them properly. Code Visualization becomes necessary, that it, it needs to manage the quality properties through the extract from the Code Visualization [1].

This paper describes an extracting method for factors of performance degradation among quality property requirements through Code Visualization. Therefore, it is organized as follows: Chapter 2 describes Code Visualization as related work, Chapter 3 shows to detect and apply factors harmful to performances, and Chapter 4 refers to Conclusion and future research.

2. Related Work: Code Visualization

Essential are software development and management, process, test automation, and quality certification, etc. for successful software development and management. However, resources and professional personnel are too lack to carry out such work and Code Visualization is recommended as a technique to improve the efficiency for maintenance and quality management [1-3]. Figure 1 shows one example of code visualization.

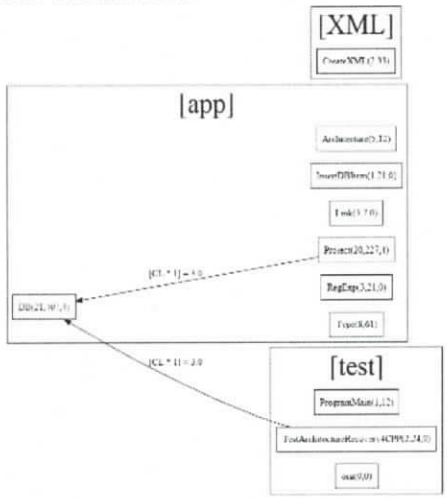

Figure 1.one example of Code Visualization

3. The Extraction Method for Performance Improvement

The degradation factors include environmental problems such as lack of memory, problems about Java virtual machine, etc. according to 2005 Information System Management Guidance [4], but 33% of the degraded SW performance is occupied by SW design and source code occurs. Therefore, a method is required to extract elements of SW performance degradation.

3.1 degradation factor extraction method

Figure 2 represents a method for extraction a degradation factor by Code Visualization.

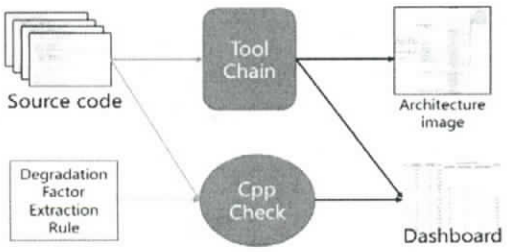

Figure 2. System Configuration

If you enter an existing source code in the Tool Chain, you can analyze the source and extract the quality properties fit in the architecture image Dashboard. Also, if you input the source code and the extraction rule about performance degradation factors in Cppcheck [5], you can output violations in the Dashboard.

Table 1.degradation factor extraction rule

Loop	[a-z,A-Z,_]([a-z,A-Z,_,0-9])* \&lt ; ([0-9])+ ; [a-z,A-Z,_]([a-z,A-Z,_,0-9])* \+\+
Control Statement	[a-z,A-Z,_]([a-z,A-Z,_,0-9])* \< ([0-9])+ ; [a-z,A-Z,_]([a-z,A-Z,_,0-9])* \+\+ \) \{ if \(
	if \(([a-z,A-Z,_,0-9,\b,\s,\+,-,\/,*,\%%,-\>,\>,\<,\[,\]])* == ([0-9])+ \)

Table1 is a regular representation of a performance degradation factor pattern.

3.2 performance degradation factors

Table 2.degradation factor example

Loop	int sum =0;
	for(i = 0 ; i<1000; i++){
	sum += array[i];}

Control Statement	for (i=0;i<1000 ;i++){
	if(i & 0x01){
	do_odd(i);
	}else{
	do_even(i);
	}
	}

	if(a==1){
	}else if(a==2){
	}else if(a==3){
	}else if(a==4){
	}.....

Table 2 shows an example of a SW performance degradation factor.

Static Analysis Tools : CPPCheck

Style Violation	Warning Violation	Performance Violation
203	58	1
178	58	1
173	58	0
143	57	0
142	57	0
97	55	0
93	55	0
93	44	0
65	44	0
65	44	171
64	44	83
64	44	43
62	37	36

Figure 3. Result of Dashboard

Figure 3 shows the results of performance degradation factor

If a performance degradation factor extraction rule is applied, it can show the number of violations like Fig. 3, and if the number is clicked, it can indicate the location of the violation (code line).

4. Conclusion

This study showed a simple pattern for reducing SW performance, and it is possible to extract a pattern to degrade its application and performance in the tool chain by a regular representation for a pattern rule. This can lead a programmer to improve a poor practice of a system performance and the existing system. The future study will be about finding and refactoring a lot of performance degradation factors, and improving performances.

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] Nipa SW Engineering Center, "SOFTWARE ENGINEERING WHITE BOOK: KOREA 2013", Nipa, KOR, 2013.

[2] Geon-Hee Kang, Keun Sang Yi, Dong Ho Kim, Jun Sun Hwang, Young Soo Kim, Young B. Park, R. Young Chul Kim, "A Practical Study on Tool Chain for Code Static Analysis on Procedural Language", KCC2014, KIISE, pp. 559-561, 2014.

[3] Bo Kyung Park, Ha Eun Kwon, Hyeo Seok Yang, So Young Moon, Young Soo Kim, R. Young Chul Kim, "A Study on Tool-Chain for statically analyzing Object Oriented Code", KCC2014, KIISE, pp. 559-561, 2014.

[4] NIA, "Guideline for Performance Management of information System", NIA, KOR, 2005.

[5] http://cppcheck.sourceforge.net/