Vol. 2 No. 2

Advanced Engineering and ICT-Convergence Proceedings (AEICP)

3rd International Conference on Advanced Engineering and ICT-Convergence (ICAEIC-2019)

Jeju International University, Jeju Island, Korea
July 11-13, 2019

Organized by
ICT-Advanced Engineering Society

Sponsered by
Jeju International University

ISSN : 2635-4586

Date of Printing: July 05, 2019
Date of Publishing: July 08, 2019

Publisher: Seongsoo Cho
Editor: Bhanu Shrestha

Publication:
ICT-Advanced Engineering Society
Bima Build. #525, 20 Kwangwoon-ro, Nowon-gu, Seoul, Korea (01897)
info@ictaes.org; +82-2-940-8626 / 8637
Website: http://www.ictaes.org

Registration No.: 25100-2018-000027

3rd International Conference on

Advanced Engineering and ICT-Convergence 2019 (ICAEIC-2019)

July 11-13, 2019

Jeju International University, Jeju, Korea

Organized by

ICT –Advanced Engineering Society, Seoul, Korea (ICT-AES)
Bima Build. #525, 20 Kwangwoon-ro, Nowon-gu, Seoul, Korea (01897)
Email: info@ictaes.org, Tel.: +82-2-940-8626 / 8637

Sponsored by

Jeju International University, Jeju Island, Korea
2870, 5.16-ro, Jeju-si, Jeju-do, Republic of Korea (63309)
Tel +82-064-754-0200, Fax +82-064-702-0933

ICAEIC-2019-0739

Selecting Facial Features for Machine Learning 13

(Jinho Han)

ICAEIC-2019-0792

The Impact of Technologies in Management Education of Nepal 16

(Mithlesh Kumar Jha, Surendra Shrestha, Subarna Shakya)

ICAEIC-2019-0731

Analysis of Privacy Disclosure Risk in CPS 19

(Min Kyoung Sung, Yong Sung Kim)

ICAEIC-2019-0714

A Study of a Medical CPS Standard Evaluation Model: MCPSSE 22

(Ju-Yuck Han, Jung-Hwan Park, Ki-Won Song)

ICAEIC-2019-0740

A Study on Development of Test Automation Framework for a Medical Software 25

(Kidu Kim, Kwanglak Jung, Gwanghun Kim)

ICAEIC-2019-0782

Software Visualization Approach for Performance Measurement of Object-Oriented Code based on Cyber-Physical Systems (CPS) Software 28

(Bo Kyung Park, Geun-Hee Kang, R. Young Chul Kim)

ICAEIC-2019-0795

On the Construction of Virtual Verification System Utilizing Unstructured Data for Reliability/Safety of Autonomous Driving Vehicles 31

(Kim Tong Hyun, Kim Young Min)

ICAEIC-2019-0741

White-Box Testing of Convolutional Neural Network 56

(Hyung-Ho Kim)

ICAEIC-2019-0725

An Analysis of Modified Fibonacci Backoff Algorithm for 802.11 DCF 59

(Sangyeon Kim, Sungwook Ko, Kangwoo Lee, Younsoon Shin)

ICAEIC-2019-0786

Dwindling Stage of Postal Services in an Underdeveloped Country 63

(Robert Khanal, Surendra Shrestha)

ICAEIC-2019-0768

A Study on Timing Issue Analysis of Cyber-Physical Systems 66

(Joon-Ik Kong)

ICAEIC-2019-0775

Emergent Failure Effects on Dependability of Cyber-Physical Systems 69

(Taihyong Yim)

ICAEIC-2019-0777

Requirements for Scalable CPS of Smart City 72

(Joonho Kwak)

ICAEIC-2019-0783

Performance Measurement of Procedural Code for CPS Multiple-Joint Robotics Simulator 75

(Bo Kyung Park, Geun-Hee Kang, R. Young Chul Kim)

Software Visualization Approach for Performance Measurement of Object-Oriented Code based on Cyber-Physical Systems (CPS) Software

Bo Kyung Park [1], Geun-Hee Kang [2], R. Young Chul Kim[3*],

[1]*SELab, Dept. of Software and Communications Engineering, Hongik University, Sejong Campus*
[2]*Telecommunications Technology Association*
[3]*SELab, Dept. of Software and Communications Engineering, Hongik University, Sejong Campus*

park@selab.hongik.ac.kr[1], lustiness_hee@tta.or.kr[2], bob@hongik.ac.kr[3]

Abstract

The Cyber-Physical System (CPS) is autonomously performed by a number of heterogeneous systems combined over the network. CPS integrates large-scale systems, and each system controls independently. This is a typical distributed information system structure. So, CPS can control complex and large data of the real world. However, in this environment, CPS requires a variety of functions and high performance. CPS SW has a huge source code and a high complexity. Therefore, SW based on CPS does not guarantee performance or quality. To solve these problems, this paper proposes a performance measurement mechanism for the CPS-based Java code. This method can identify which part has a problem.

Keywords: *CPS; SW Performance; Code Visualization; Refactoring*.

1. Introduction

Cyber-Physical Systems (CPS) is an essential technology for the fourth industrial revolution such as smart factory and smart home [1]. CPS is acquiring, processing, computing, and analyzing physical world information. This result applies to the physical world through the actuator system. In other words, CPS is an extended concept of the existing embedded system. These embedded systems often have limited resources such as available power, memory, and performance. In this case, the embedded system can cause many operational problems. These embedded systems require various functions and high performance. For this reason, software becomes more complex and large source code. As a result, SW based on CPS does not guarantee performance or quality. To solve this problem, this paper proposes a performance measurement mechanism for Java code based on CPS. The procedure is as follows: We extract information from the source code to identify the internal structure of the CPS SW code. We define a module to represent the internal structure. We measure the coupling indicators of SW. Define performance degradation factor patterns. We measure the quantitative value of SW performance through profilers. Finally, we identify and improve the

ISSN : 2635-4586

problem with the software. We expect to improve the performance and quality of CPS SW. The composition of this paper is as follows: Chapter 2 introduces the related study. Chapter 3 describes the performance measurement mechanism of SW code (java) based on CPS. Chapter 4 shows case studies. Chapter 5 refers to conclusions and future studies.

2. Related Works

Code visualization visualizes the internal structure of the source code. Code visualization improves software quality based on coupling and cohesion [2]. It constructs tool-chains using several open-source tools. Tool-chain consists of a source navigator, SQLite, and view composer (Graphviz). Source Navigator is a parser that analyzes the source code. SQLite saves the analyzed data. View Composer generates an image of the analyzed data.

3. Performance Measurement Mechanism for CPS-based Java Codes

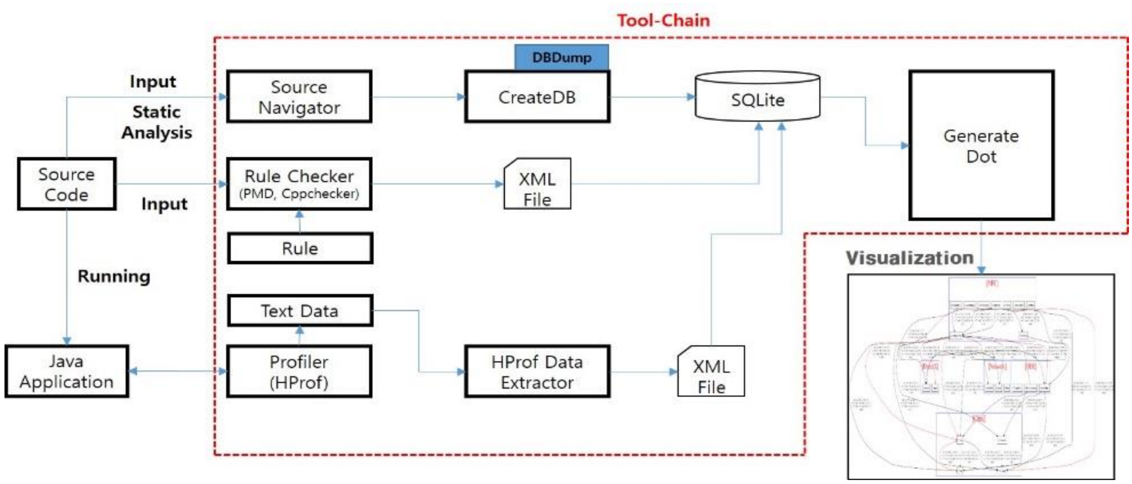

Figure 1. Performance Measurement Mechanism for CPS-based Java Codes

Figure 1 shows a performance measurement mechanism for cps based java code. This mechanism added tool-chain, rulechecker (PMD), Profiler (Hprop) [3], and HprofDataExtractor. Tool-Chain applied the existing code visualization method [2, 4]. RuleChecker extracts bad patterns of software performance [5]. The Profiler dynamically analyzes the CPS SW. The HprofDataExtractor refined the data from the Profiler into XML data.

4. Case Study

We applied some of the robot control and simulation programs (Couplic_sample project). Figure 2 is the detailed class view with performance indicators. The class name is displayed at the bottom of the figure (SampleDataClass). The Detailed Class View is divided into before and after clusters. Each cluster represents the method name, execution frequency, and execution time in the class. The arrows connect the methods of each cluster. The arrow value is the changed execution time. For example, there are a getB method of before and a getB method of after. The value between these methods is -2347.4297 (ms). This value indicates that the speed has improved since the refactoring.

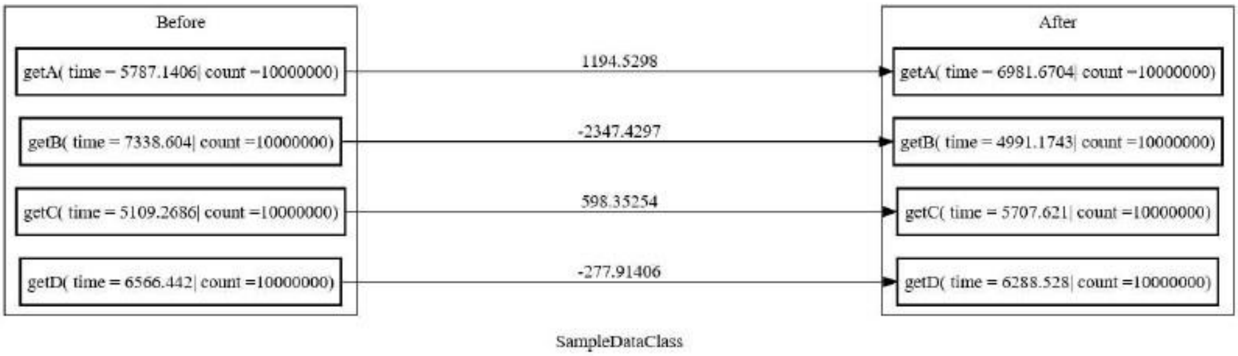

Figure 2. The Detailed Class View with Performance Indicators

5. Conclusion

In this paper, we propose a performance measurement mechanism for the high quality of CPS SW. We extracted the coupling and performance degradation elements through code visualization. Coupling increases the reusability of the CPS SW. Performance degradation is a factor that violates SW performance. We also performed dynamic analysis through the Java performance measurement tool (HProf). This can extract information on the execution speed of the method and the execution frequency of the method, and cannot be obtained from the static analysis. We could see the change of sw performance through the improvement of coupling. The change value of software performance is as follows: The speed 0.59%↑, 53.26%↑, 73.25%↓, 4.98%↓. Performance is improved when the stamp or external coupling is improved. Therefore, we recommend refactoring. Through code visualization, many developers and stakeholders can easily understand the software's quality indicators and structure. In the future, we will find out the performance impairment factors proposed in this paper and the dependent performance problems of each code.

Acknowledgement

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [201900340001, Development of Micro-service IoTWare Framework for Micro IoT Devices] and this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A3B03035421)

References

[1] Rugunathan, R., Dionisio, N., & Mark, K. (2018). Cyber-Physical Systems. Acorn.

[2] Bokyung. Park, Haeun, Kwon, Hyeoseok, Yang, So Young Moon, Young Soo Kim, R. Young Chul Kim. (2014, June). A Study on Tool-Chain for statically analyzing object oriented code (pp. 463-465). KIISE.

[3] Hporf: A Heap/CPU Profiling Tool. https://docs.oracle.com/javase/7/docs/technotes/samples/hprof.html

[4] Geon-Hee Kang, Keun Sang Yi, Dong Ho Kim, Jun Sun Hwang, Young Soo Kim, R. Young Chul Kim. (2014. June). A Practical Study on Tool Chain for Code Static Analysis on Procedural Language (pp559-561). KIISE.

[5] PMD. https://pmd.github.io