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

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

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

Cyber-Physical Systems (CPS) deals with complexity, uncertainty, and unpredictability through an abstract model. However, CPS simulations are simulated simultaneously for many embedded systems. Therefore, the code size of the CPS-based software will increase and the complexity will increase. This CPS Software does not guarantee performance or quality. For the efficient performance of CPS SW, we propose a method of visualizing performance measurement of the CPS-based C/C++ code. We expect the performance and quality improvement of CPS SW.

Keywords: *CPS; SW Performance Measurement; Code Visualization; Refactoring; SW Quality Improvement.*

1. Introduction

Existing control systems evolve from simple structures to complex structures of numerous computing units. So, many systems cause errors and problems. CPS deals with complexity, uncertainty, and unpredictability through an abstract model [1]. The CPS model needs to verify the software using simulation technology. However, CPS simulations are simulated simultaneously for many embedded systems. Therefore, the code size of the CPS-based software will increase and the complexity will increase. This CPS Software does not guarantee performance or quality. Therefore, a mechanism is needed to ensure the performance of CPS Software. For the efficient performance of CPS SW, we propose a method of visualizing performance measurement of the CPS-based C/C++ code. This method extracts the complexity of CPS-based C/C++ codes through code visualization. We select the most complex modules, and we refectories them. We can improve the CPS-based C/C++ code. The proposed method applies complexity and performance to CPS SW. So we expect the performance and quality improvement of CPS SW. The composition of this paper is as follows: Chapter 2 describes the related research. Chapter 3 describes the performance visualization of the CPS-based C/C++ code. In chapter 4, we explain the results before and after refactoring. Chapter 5 refers to conclusions

ISSN : 2635-4586

and further research.

2. Related Works

In CPS SW, modeling and SW validation techniques measure SW's performance. This technology ensures the simplification and high reliability of the system. Modeling should consider a computational model to represent the system that we are currently developing. This model is divided into a discrete model and a continuous model. The discrete models model a digitized software system. The continuous model models model systems with analog properties, such as the real world. However, the IT convergence industry has emerged recently. The hybrid model is a fusion of discrete and continuous models [2].

3. Performance Measurement Mechanism for CPS-based C/C++ Codes

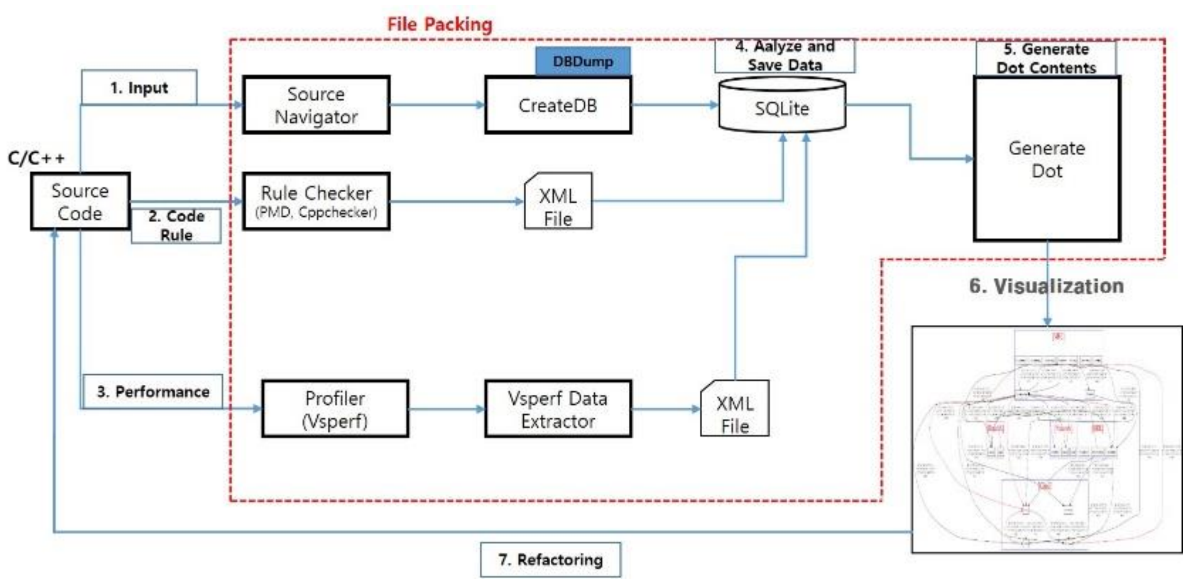

Figure 1. Performance Measurement Mechanism for CPS-based C/C++ Codes

This paper extracts the core module of the CPS code through code visualization. It measures and improves the performance of core models. These processes are as follows. In the first step, we input the target source code into the performance measurement system. We extract the structure and complexity of the code. We analyzed the results and selected the highest complexity module. This method can improve the SW performance by entering the entire code. However, for efficient performance, we verify the highest complexity module. The selected module performs the refactoring. The refactoring process extracts a pattern of performance degradation factors through RuleChecker. The extracted data is stored in SQLite as an XML file. We use VsPerf for dynamic analysis of c/c++ codes [3]. The data extracted from VsPerf is refined into XML data using VsPerfDataExtractor. We store data in SQLite. Finally, we extract performance visualization graphs. The information in this graph is as follows: C/C++ code data in violation of performance, Existing CPS SW structural information (Coupling), Memory usage information in the module, etc.

4. Case Study

Figure 2 is a performance measurement result of Multiple-Joint Robot Simulator. This performance result is a measure of performance by performing a refactoring of the CPS C/C++ code based on the baseline. Performance changes are divided into before and after clusters. In the most complex CSimulationView, each cluster represents the name, number of operations, and performance time of the methods inside the class. The arrows link the methods of each cluster. The figure above the arrow is the changed performance time. The performance measurement between Before(①) and After(②) is "-45.21". This is 45.21(ms) faster than before refactoring. Before(②) and After(③) is "-23269.67". This is 23269.67(ms) faster. In the third refactoring, Before(③) and After(④) is "-68.66". This means that the 68.66 (ms) speed is faster.

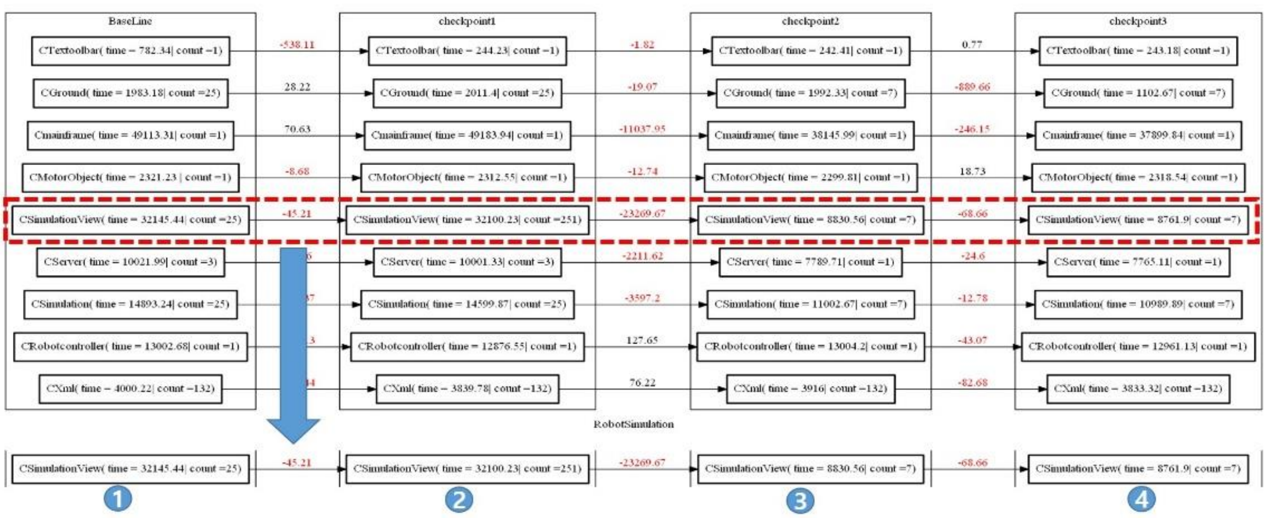

Figure 2. Performance Measurement Result of Multiple-Joint Robot Simulator (RobotSimulation)

5. Conclusion

Simulation techniques are used to verify the CPS. However, CPS simulations are simulated simultaneously for various embedded systems. The code of the CPS-based software becomes enormous and the complexity increases. These CPS SWs do not guarantee performance or quality. This paper proposes a performance measurement visualization of CPS-based C / C ++ code for efficient implementation of CPS SW. We extract the complexity of the CPS SW through code visualization. We choose the highest complexity module (core algorithm). Finally, we measure performance. Developers and stakeholders can also easily identify memory performance indicators and structures for CPS SW. In the future, we find patterns of power-consuming source codes and apply them to visualization.

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] H.S. Son, W.Y. Kim, I.G. Jeon, H.Y. Lee, J.H. Jeon, R. Youngchul Kim (2011. Feb). A Method of Representing Topographic Environment Data through SEDRIS in Cyber Physical System, the Korea Society for Simulation Magazine.

[2] W.T. Kim, I.G. Jeon, S.H. Lee, S.M. Park (2010, July). CPS Technical Trends. NIPA Weekly Technical Trends.

[3] VSPerf. https://docs.microsoft.com/ko-kr/visualstudoi/profiling/veperf.