An Automated Test Method for Robot Platform and Its Components

Jae-Hee Lim[1], Suk-Hoon Song[1], Jung-Rye Son[1], Tae-Yong Kuc[2]
, Hong-Seong Park[3], Hong-Seok Kim[4]
[1,2] School of Information and Communication, Sung Kyun Kwan University, Suwon 440-746 Korea
[3]Department of Electronic and Telecommunication Engineering, Kangwon National University
[4]Korea Institute of Industrial Technology, Ansan 1271-18 Korea
[1]homze007@skku.edu, [2]tykuc@yurim.skku.ac.kr

Abstract

This paper presents a hierarchical test model and automated test framework for robot software components of RTC(Robot Technology Component) combined with hardware module. The hierarchical test model consists of three levels of testing based on V-model : unit test, integration test, and system test. The automated test framework incorporates four components of test data generation, test manager, test execution, and test monitoring. The proposed testing model and its automation framework is proven to be efficient for testing of developed robotic software components in terms of time and cost. The feasibility and effectiveness of proposed architecture for robot components testing are illustrated through an application example along with embedded robotic testbed equipped with range sensor hardware and its software component modeled as an RTC.

Keywords: *Robot Software Component Testing, Robot Hardware Testing, Hierarchical Test Model, Automated Testing System, Robotics.*

1. Introduction

As robotic systems are getting more complicated and their application area broaden, the researches on development and standardization of robotic software platform has been reported recently. The standardization of robot software platform is aiming at more efficient customization and manufacturing of robotic products than without. To this end, the component based development approach has been used for generation of RTC(Robot Technology Component) and OPRoS(Open Platform for Robotic Services)[1,2]. However, for the component based robotic software and its platform to be applicable in common, the reliability on performance of software components and their conformity with and portability to different robotic systems have to be insured. In order to achieve these requirements, it is essential that the usability and operability of robot software components are tested during their development process. Nevertheless, few research results on testing of robotic software components have been reported so far.

In view of this, a hierarchical test model and automated test framework is proposed for robot software components of RTC(Robot Technology Component) combined with hardware modules. Based on V-model, the hierarchical test model consists of three levels of testing: unit test, integration test, and system test. The automated test framework incorporates four components of test data generation, test manager, test execution, and test monitoring. The framework allows us to more easily perform robot component test by applying an available testing technique corresponding to its test object and test level defined. Together with the proposed test model, it also provides user interface, test engine, test resource repository, etc. This paper implements a test-bed for testing of the proposed system and verifies its efficacy through a series of experiments.

2. Test Model and Test Framework

2.1. Test Model Structure

Test model is based on RTC, the standard robot software component of OMG. Fig. 1 shows the structure of test model in conformity with RTC and its application example to an ultrasonic range finder sensor.

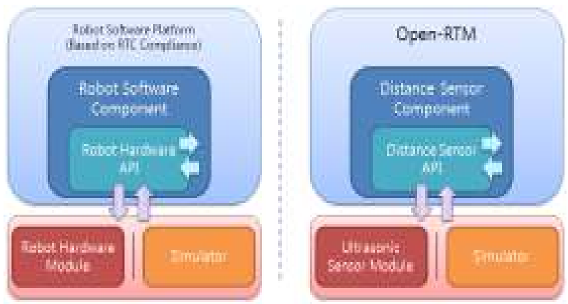

Figure 1. Test Model Structure(left) and Its Application Example(right)

The overall robot software test system is composed of robot software platform, robot software component, robot hardware API, robot hardware module, and simulator. Robot hardware module is hardware part of robot and simulator is a virtual robot hardware platform which can accommodate robot hardware API in place of robot hardware module. Robot hardware API provides common parts of robot hardware modules in the form of prototype function. The body of robot hardware API is defined as library or DLL in accordance with robot hardware module.

Fig.2 shows an implementation of test model for range finder sensor component conformed with the standard robot software component RTC.

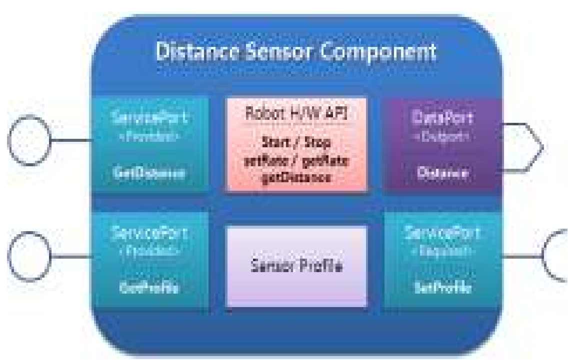

Figure 2. Robot Software Component Test Model for Range Sensor

Robot software components complying with RTC standard communicate with each other through ports. In Fig.2, the range finder sensor component possesses 1 data port for range data output and 3 service ports for transmission of internally executed functions to other component. Three internal functions include GetDistance for distance value and setProfile/getProfile for transmission of component profile.

2.1. Outline of Hierarchical Testing Procedure

Since robot software component operates tightly coupled with its corresponding robot hardware module, robot component testing procedure needs to accommodate hardware and its interface as well as robot software component. In view of this, a hierarchical testing procedure is set up in this paper for testing of robot component conformed to RTC. Fig. 3 shows the proposed hierarchical testing procedure model which includes three levels of testing : unit testing, integration testing, and system testing for robot component. The three testing levels correspond to hardware testing, hardware API testing, and composite software component testing, respectively. That is, in Fig. 3, hardware module is considered as a basic unit for hierarchical testing of robotic software component.

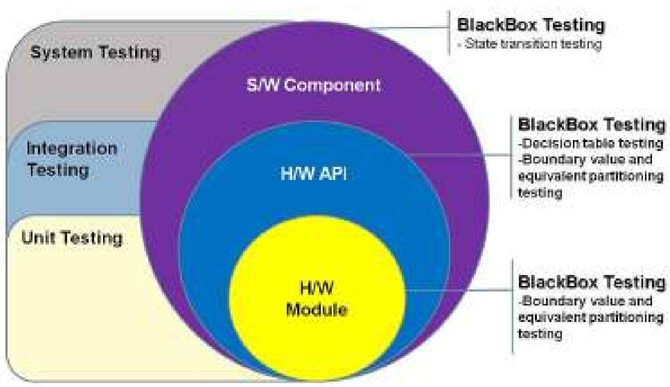

Figure 3. Hierarchical Testing Procedure for Robot Software Component

After unit testing for validation of hardware module, the interoperability of hardware module and software component is checked by performing integration testing. In this step, robot hardware API is tested for performance index of functionality by using test cases derived from black box testing techniques such as boundary value analysis, equal partitioning test, decision table testing, etc. The performance index of functionality includes completeness of function realization, correctness of data, compatibility of data, etc. In the final step of system testing, a series of operations are tested for software component which are specified in the document of software component requirement. The performance index for system testing consists of functionality(compatibility of document, exactness of state transition, correctness of data), maintenance(comprehensibility of cause of defect), portability(functional alternativeness), etc. The testing techniques of boundary value analysis, equal partitioning testing, state transition testing, etc. are used for system testing of robot software component.

2.3. Testing Automation Framework

Although robotic software testing is crucial for development and application of standard robotic software platform and its components, lack of manpower and resources and insufficient time induces difficulty in provision of systematic testing process during robotic software development and application. In order to reduce time and cost for testing of robotic software, it is important to develop standard test model and automatic testing framework which provides systematic testing procedure.

To this end, a testing automation framework is designed for testing of robotic software component based on the hierarchical test model presented in the previous subsection. Fig. 4 shows the proposed testing framework consisting of user interface, test engine, and test resource repository. User interface implements three levels of testing hierarchy : robot hardware module test UI for document and performance test, API test UI for interface test, and robot software component UI for functional test. Test engine includes test manager, test data generation unit, test execution unit, and test monitoring unit. Test resource repository provides materials necessary for execution of test process to test engine such as test code, test plan, test case, metrics of performance measurement, etc. It is also illustrated in Fig. 4 that test engine can operate linked with robot hardware testbed or robotic simulator provided.

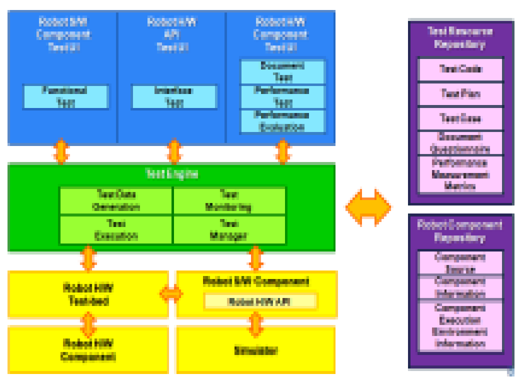

Figure 4. Shcematic of Testing Automation Framework

Fig. 5 demonstrates the operation of test engine by showing the interplay between function blocks of inner and outer parts.

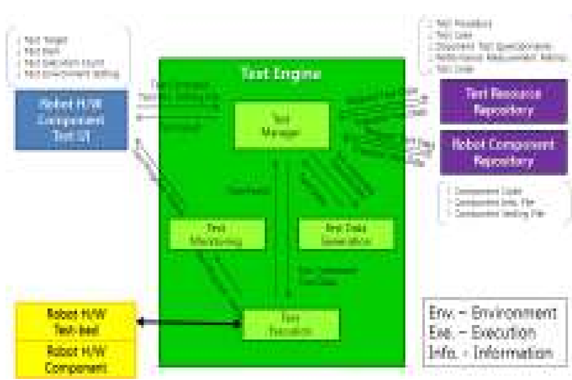

Figure 5. Operation Flow of Test Engine

2.4. An Application Example

The proposed test model and its test framework are applied for testing of range sensor component mounted on a robot hardware testbed. Fig. 6 shows the details of test environment operating coupled with the test framework in Fig. 7.

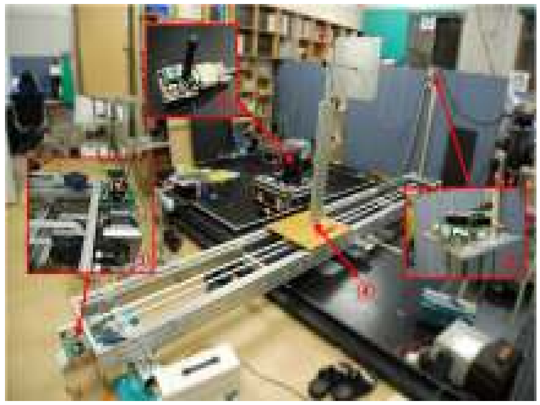

Figure 6. Testbed for Evaluation of Range Sensor Component

Testbed system hardware consists of wireless communication station(①), ultrasonic range sensor module(②), motor drive unit(③), and target object carrier(④). The wireless communication station connects testbed system to the main PC of robot test engine which controls the ultrasonic sensor and motor drive unit for synchronization of testing procedure.

Fig. 7 shows the block diagram of overall experimental setup including the main PC working as test agency.

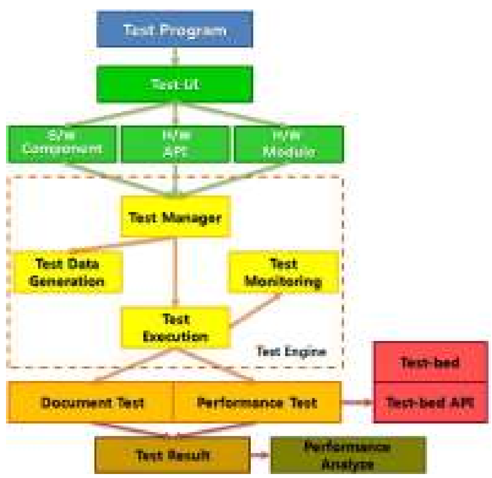

Figure 7. Block Diagram of Overall Testing System

Figure 8.UI of Overall Testing System

Fig. 8 shows user interface for testing framework implemented using MFC. In the figure, testing framework supports test case generation along with testing target, test execution, test result display, etc.

Test case refers to performance indices shown in Fig. 9, where range sensor example is given. Boundary value analysis and equivalent division methods are used for the derived test cases shown in Table 1 and 2.

Figure 9. Performance Indices for Evaluation of Range Sensor Module

Table 1. The Test Case for Data Correctness of Range Sensor Module by Equivalent Division Method

test case	1	2	3
distance value	2	150	400
range	distance < 3	3 <=distance<= 300	distance > 300
expected result	Timeout	distance value	Timeout

	(unmeasurable)		(unmeasurable)

Table 2. Test Case for Data Correctness of Range Sensor Module by Boundary Value Method

test case	1	2	3	4
distance value	2	3	300	301
expected result	Timeout (unmeasurable)	distance value	distance value	Timeout (unmeasurable)

Fig. 10 demonstrates experiment result for correctness test of range sensor data by using ultrasonic sensor module SRF-04. Similarly, Fig. 11 shows the experiment result for fault tolerance test of range sensor data by using the same ultrasonic sensor module.

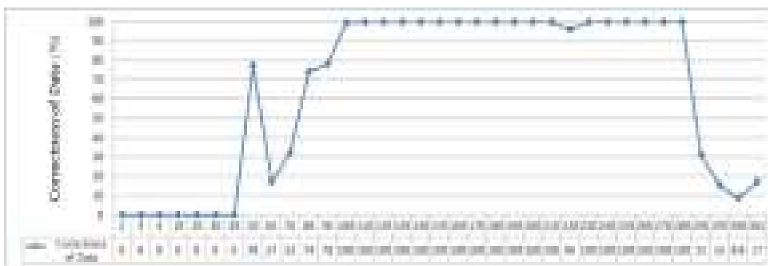

Figure 10. Correctness of Data for Ultrasonic Sensor Module(SRF-04)

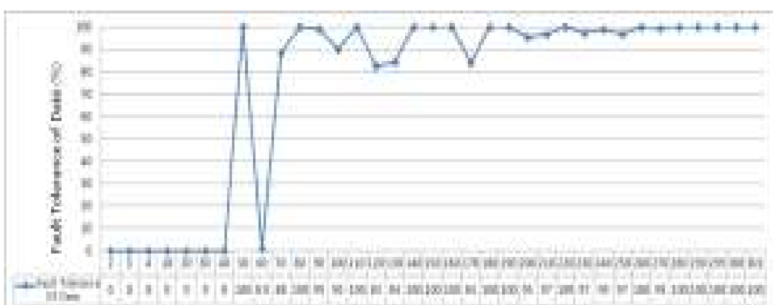

Figure 11. Fault Tolerance of Data for Ultrasonic Sensor Module(SRF-04)

The ultrasonic sensor module used in the experiment covers a wide range of 3cm ~ 300cm. Fig. 10 plots the analyzed data correctness of ultrasonic sensor based on experiment result. In the figure, data correctness was defined as equation(1).

$$Data\,Correctness = \frac{number\ of\ test\ success}{number\ of\ trials} \times 100 \quad (1)$$

Where test success means the case that its measurement value remains within normalized error bound.

In the graph, it is found that the ultrasonic sensor SRF-04 module operates correctly in the range 100~280 cm. On the other hand, data correctness decreases notably at the distance less than 80cm or larger than 290cm. The experiment results show that performance of ultrasonic sensor module SRF-04 does not agree with the product specification for operation range of 3cm~300cm.

Fig. 11 plots fault tolerance rate of ultrasonic sensor tested. In the figure, fault tolerance is defined as the % rate of data within permissible error bound among data corresponding to test success. Fault tolerance rate is computed by using the following equation.

$$Fault\ Tolerance = \frac{number\ of\ \ fault\ tolerance\ data}{number\ of\ test\ success} \times 100 \qquad (2)$$

Where fault is defined as the test result whose measurement value does not agree with the expected one. Similarly, the number of fault tolerance is defined as among the faults the number of faults within allowable error bound. Hence, the fault tolerance provides a reliability measure of sensor data for successful test.

In this experiment, a series of tests and its results has been generated demonstrating the effectiveness of proposed test model and automatic test framework for robot software component of range sensor module. Experiment verifies that the use of hierarchical test model and automatic test framework supports efficient testing of robotic software components in terms of time and cost by generating various test cases in systematic way.

3. Conclusion and Further Research

As personal robot system and robotic apparatus spread fast in various fields, the higher functionality and better performance are in great demand nowadays. In addition, the guaranteed reliability of interactive robot is of special importance in user's real life, since its defects might cause fatal damage to man and economic loss. In this respect, the performance and safety test of robotic software component together with robot hardware module is crucial for stable robotic interaction with human and its working environment.

Considering the importance of robotic software validation, a hierarchical testing model and its testing automation framework are developed for RTC compatible robotic software components. The effectiveness of proposed testing model is demonstrated through a series of real experiments for an embedded robotic testbed equipped with ultrasonic sensor module as range finder. Experiment results show that the proposed hierarchical testing model and testing automation framework provide an efficient way of robotic software testing in terms of time and cost. It is expected that the developed testing model and its automation framework is applicable to various standard robotic software components in real life as well as in development stage.

Acknowledgments

This work was supported by Knowledge and Economics Department of Korea Government through OPRoS(Open Platform for Robotic Services) project and NRHP(National Robot Hardware Platform) Technology project.

References

[1] OMG, The Robotic Technology Component Specification, 2007.

[2] OPRoS, Open Platform for Robotic Services Specification, 2008.

[3] W. Kwon etc., Software Testing Service for Developer, STA, 2008(in Korean).

[4] S. Hong, Embedded Software Testing, SW Insight Report, 2008(in Korean)

[5] TTA S/W Test Center, S/W Quality Test System and Technology Survey, 2004(in Korean).

[6] ITEA, Guideline for V&V Real-Time Embedded Software Systems(DESS), 2001

[7] ETRI, Construction of Embedded System Development Framework, 2006(in Korean).

[8] OASIS, ebXML Test Framework Committee Specification V1.0, March 2003.

[9] The Open Group, TETware White Paper.

[10] STA, TPMS Product Instruction Manual, 2007.

[11] H. Martin and L.du Bousquet, "Automatic Test Generation for Java-Card Applets", Lecture notes in computer science, Vol 2041, 2001, pp.121-136.

[12] Tan L., Kim J., Sokolsky O., and Lee I., "Model-based Testing and Monitoring for Hybrid Embedded Systems", Information Reuse and Integration, 2004, pp.487-492.

[13] Jervan G., Eles P., Peng Z., "A Hierarchical Test Generation Technique for Embedded Systems", Proc. Electronic Circuits and Systems Conference, 1999, pp.21-24.

[14] Chanhyun Baek, Joongsoon Jang, Gihyun Jung, Kyunghee choi, and Seungkyu Park, "A Case Study of Black-Box Testing for Embedded Software using Test Automation Tool", Journal of Computer Science, 2007, pp144-148.

[15] ISO/IEC TR 9126, Software Engineering-Product Quality-Part 1,2,3,4, 2005.

Authors

Jae-Hee Lim received the B.S. degree in Computer Science from DanKook University, Korea in 2007 and the M.S. degree in the Department of Electrical and Computer Engineering from SungKyunKwan University, Korea in 2009. Since June 2009, she has been working at Obigo Korea, where she is currently a Test Engineer. Her research interests include test automation, test methodology, widget, web-platform.

Suk-Hoon Song received the B.S. degree in Computer Engineering from Kyungwon University, Korea in 2008 and the M.S. degrees from Sung Kyun Kwan University of Electrical and Computer Engineering, Korea in 2010. His research interests include Robot Vision, software-hardware testing, and mobile robot navigation.

Jung-Rye Son received the B.S. degree in Computer Engineering from Seokyeong University, Korea in 2009. She is currently working toward the M.S. degree at the Department of Electrical and Computer Engineering, Sung Kyun Kwan University, Suwon, Korea. Her research interests include robot vision, intelligent control and robotics, robot software platform.

Tae-Yong Kuc received the B.S. degree in Control and Instrumentation Engineering from Seoul National University, Korea in 1988 and the M.S. and Ph.D. degrees from Pohang University of Science and Technology, Korea in 1990 and 1993, respectively. From April to August 1993, he worked as Chief Research Engineer at Precision Machinery Institute of Samsung Aerospace Company and from September 1993 to February 1995 as Senior Lecturer in the Department of Electrical Engineering, Mokpo National University, Korea. Since March 1995, he has been with the School of Information and Communication Engineering at Sung Kyun Kwan University, Suwon, Korea, where he is currently a Professor. His research interests include intelligent robotics, adaptive and learning control, intelligent character animation, and sensor-data fusion for computer-aided control systems.

Hong-Seong Park was born in Korea in 1961. He received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1983, 1986, and 1992, respectively. Since 1992, he has been with Department of Electrical and Electronics Engineering, Kangwon National University, Korea. His research interests include performance analysis of communication networks and mobile networks, network-based control systems, and SW engineering including testing.

Hong-Seok Kim received B.S., M.S. and Ph.D. degrees in Electrical Engineering from Seoul National University in 1980, 1983 and 1990, respectively. He has been with the Korea Institute of Industrial Technology (KITECH) since April 1991, as a Chief Research Engineer. He was in charge of managing the "Next-generation Growth Engines" national R&D program in robotics industry, as the Managing Director of the Center for Intelligent Robot (October 2006~June 2009) and is currently the Senior Director of the Division for Technology Convergence Research at KITECH since May 2009. His interests include controller design, robot hardware platform technology, and Services Science focused on service robotics.