International Conference on Control, Automation and Systems 2007
Oct. 17-20, 2007 in COEX, Seoul, Korea

Software Testing for Intelligent Robots

Yun Koo Chung[1] and Sun-Myung Hwang[2]

[1] Human Robot Interaction Research Team, ETRI, Korea
(Tel : +82-42-860-5032; E-mail: ykchung@etri.re.kr)
[2] Department of Computer Engineering, Daejeon University, Korea
(Tel : +82-42-280-2544; E-mail: sunhwang@dju.ac.kr)

Abstract: Software plays a key part in the intelligent robots. Recently intelligent service robots began to be commercialized, by which the testing and evaluation of robots became an important method for quality assurance of intelligent robots. This paper focuses on the testing technology for software components of intelligent robots. The test case design using scenario is an effective methodology to cover all user requirements. We are studying the existing software testing standards of ISO and try to adapt the test specifications to the robots[2]. Software of Intelligent robots consists of middleware, communication module, navigation module, voice and image processing components, and various kinds of application service modules. This paper talks the process of testing, testing items, and evaluation methods for software quality assurance. Testing process considers requirement analysis, design, planning, testing and reporting. Testing items are defined in ISO standard for software testing. We are going to selectively choose minimum number of items to the robots. Also testing case and environment conditions are discussed for test reference.

Keywords: Software testing, Intelligent robot, testing process, testing items

1. INTRODUCTION

Network based Intelligent Service Robot technology has been developed for the past four years in Korea. This technology is also called "Ubiquitous Robot Companion (URC)." The pilot business project of URC has been started as a national project from two years ago. This business model provides new concepts for the robot business industry. During the past and current years, most of countries are developing humanoid robots whose shapes are similar to humans and whose functions are artificially intelligent. This technology is still taking a lot of years to be practically used. Robot experts say that it will need more than 10 years even though they invested more then several ten years in the past. Korean robot experts had strategically interests in another new area. They are to fuse robot technology and IT technology, by which the network based intelligent service robotic project has been created as URC project. The purpose of the URC is that the robots provide services that the users want anytime and anywhere in low price. At this time, the robot price has been at least more than several USD 10,000s. It is not easy to buy it at home and still the functions of robots are not satisfied.

By adopting the server/client systems to the robot technology, the price and the function can be dramatically saved and improved. In URC, the system consists of three parts: computing, sensing and action with a network. If we use servers for computing of robot functions, high quality of service functions can be delivered to client robots using a high speeded network. The general high speeded network for URC robots are more than 10Mbps. During the last year, Korea Telecommunication Company 'KT' leaded the pilot business project and delivered service network based URC robots to 850 apartments. Many URC servers and the high speeded networks were successfully used to provide the various IT services by URC home robots. We estimated the client robot prices as less than USD 2,000. From the first pilot business project, many problems and opinions were given to KT. We are to resolve the problems of malfunctions and problems by restricted quality testing of robot products and also improving the technology, where especially speech recognition service and automatic battery charging were mentioned.

During the year 2006, the authentication test daft was made and the most focusing was on complying test to the telecommunication protocol between the robots and the servers. The testing draft should be based on the specifications of robot standards. But this area is still on the beginning stage. This paper is to provide the guide line to be referenced on this purpose.

2. BACKGROUND OF TESTING

The year 2007 is the second year of URC home robot pilot business project. During the past years, the technology development is the most important but for commercializing the intelligent robots, the strict testing should be applied to the robot products. Otherwise, we may expect many kinds of malfunctions from the serviced robots through the pilot business. The testing is the least filter to prevent these kinds of problems.

During the last year, all URC robots produced from six companies failed to be passed in the initial

Fig. 1 URC robots

978-89-950038-6-2-98560/07/$15 ⓒICROS

authentication tests of URC robots as shown in Figure 1. Considering the pilot business and encouraging the robot companies, the second test was applied with lower specifications. Many weak points and problems were discovered by the tests. Also the user surveys were carried out after three months of usage for URC home robots. The most important problems of technologies should be improved and resolved. Therefore, we have kept the strict authentication test and also more evaluation items are added in this year.

The testing reference is based on ISO 9126 and ISO 14598 which is the software evaluation rules. About 80 items of ISO standards are too much to the robot tests. We extracted the most important 30 items among the ISO specifications for software test. The software has an important key role in intelligent robots since the intelligence is mostly implemented by computer programs and the software recently is managed in software life method. But robot software is performed with the robot hardware devices. Considering these facts, the performance testing should be more weighted in the testing evaluation of performance. Therefore, the software testing method is suitably combined with the performance tests for authentication tests.

This paper suggests that testing methods for intelligent robots are studied and the results are recommended for authentication tests of robots. The testing process and the testing methods are suggested for this purpose in this paper.

3. SOFTWARE TESTING PROCESS

The testing process consists of the following five steps: Test requirement Analysis, Test details, Testing design, Processing the tests and Report of Test results.

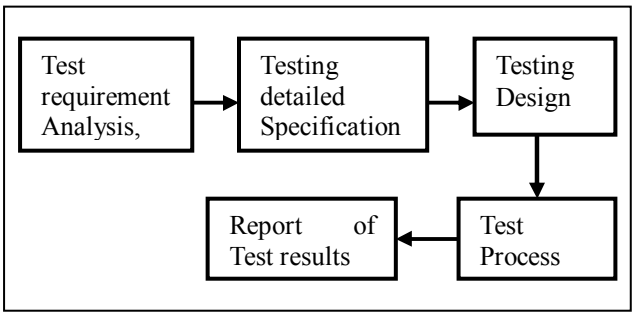

Fig. 2 Testing Process of Robot Software

The testing process of robot software is provided for users of robots. The users can be producers and users of robot products. ISO/IEC 14598-5 defines the above five steps of testing processes and activities for software quality assurance:

- Test requirement analysis: The demanding requirements of users are collected and analyzed to find the demanding functions from the users.
- Testing specifications: The detailed testing specifications are drafted from the user requirements and the product user manuals.
- Testing design: Testing plan is drafted based on testing specifications, target software, testing methods in detail.
- Testing process: Inspection, modeling, measuring and testing are performed based on the testing plan.
- Report of test results: Test results are concluded by acceptance of the test report.

In order to follow the above testing process for the robot software, the evaluation items are defined in the ISO/IEC 9126 as the main characteristics with the subcharacteristics for measuring the software.

4. SOFTWARE QUALITY CHARACTERISTICS

The software is measured according to the ISO/IEC 9126 quality characteristics with the subcharacteristics. This is called metrics. The software quality characteristics consist of the following six main characteristics with more detailed subcharacteristics.

4.1 Functionality

This measures the capability of software products according to the specifications in the user requirement analysis when the software is used under the specified conditions. The following table explains the subcharacteristics of the functionality.

Suitability	Capability to provide an appropriate set of functions for specified tasks and user objectives
Accuracy	Capability to provide the right or agreed results of effects with the needed degree of precision
Interoperability	Capability to interact with one or more specified systems
Security	Capability to protect information and data so that only authorized users or systems are accessed to them
Functional Compliance	Capability to adhere to standards related to the functionality

4.2 Reliability

It is the capability to maintain a specified level of performance when it is used under the specified conditions. The following table explains the sub characteristics of the reliability:

Maturity	Capability to avoid failures as a result of faults in the software
Fault tolerance	Capability to maintain a specified level of performance in case of software failure or infringement of its specified interface
Recoverability	Capability to re-establish a specified level of performance and recover the data directly affected in case of failure
Reliability compliance	Capability to adhere to standards related to the reliability

4.3 Usability

It is the capability to be understood, learned, used and attractive to the users under specified conditions.

Understandability	Capability to enable users to understand whether the software is suitable
Learnability	Capability to enable users to learn its application
Operability	Capability to enable users to operate and control it
Attractiveness	Capability to enable users tobe attractive to users
Usability Compliance	Capability to adhere to standards related to the usability

4.4 Efficiency

It is the capability to provide appropriate performance related to the resources used.

Time behavior	Capability to provide appropriate response and processing time and throughput rates
Resource behavior	Capability to provide appropriate amounts and types of resources
Efficiency Compliance	Capability to adhere to standards related to efficiency

4.5 Maintenability and Portability

It is the capability of software maintenance to be modified. Also portability is the capability of software to be transferred from one environment to another. Maintenability consists of analyzability, changeability, stability, testability and compliance. Portability consists of adaptability, instability, co-existence, replaceability and compliance.

5. URC ROBOT SOFTWARE COMPONENTS

The components consisting of the URC robot software is composed of HRI (Human-Robot Interaction), robot smart-action technology, and others.

5.1 HRI Technology

URC robots interact with users for user recognition or some kind of expressions, which is different from other information machinery. Robot technology requires the recognition and response of user's voice, the verbal transmission through voice composition on the information, the recognition to user's facial expression and gesture from the image signal through cameras. Chart 6 contains the contents of the human-robot interaction technology which is under development. This software module is offered by URC service, and it is developed by the aim focused on the reaction to 100 clients within one second with simultaneous contacts.

The final aim of voice cognition is to recognize the dialogue style with user's natural utterance in the noise environment. But at present, it maintains a standard of

TABLE 1 HRI TECHNOLOGY

HRI component technology	Infra-Image HRI component technology	User Recognition	User's cognition module by an quasi-organism information, facial expression at a long distance in the situation of robot's upward sight
		Gesture Recognition	User's pursuit and gesture cognition module development Caller Identification module
		Expression Recognition	Facial component detection and special feature extraction module
	Infra-Voice HRI component technology	Voice Recognition	Cognitive module of user's voice information transmitted from robot
		Voice Decomposition	Conversion module changing arbitrary sentence with composition voice
		Voice Pursuit	Exclusion module removing noises in the robot and outside noises for tracking the original sound
		Speaker Recognition	Speaker cognition module by the context speaker independent method

using the cognition apparatus of isolating languages and the cognition lexicon for sentence units considering user's natural utterance.

We have developed the voice recognition, aiming at the cognitive rate of 95% as a basis of linking language cognition, within 2 meter, in the limit of 20dB, with only once utterance, perceiving subsidiary lexicon which make up important lexicon and declinable word corresponding to the key words in URC.

Voice composition is to convert arbitrary sentence into composition voice similar to human voice. According to the various services, URC can change the output composition voice being suitable to the usage, as a basis of limitless lexicon composition. Tagged Text Processing is used for the purpose of the various outputs. That is, it modulates the speed, the volume, the pitch, chooses the various composition voices of children, men, women, and supplements the background sound or effect sound through the input of XML pattern.

User's cognition is a kind of technology that finds out the master of the robot and the master's identity by robot's analyzing the inputted facial image in camera. In order to overcome the limitation of existing, cooperative, user's cognition technology in URC, we have developed the technology that recognizes the natural, acting user with using the information of semi-Biometrics, such as height, dress color, and so forth, in ordinary life, and we have also been developing and studying user's emotion through caller's facial expression and discernment module that make the robot accept the user's order, dialogue or request spontaneously with recognizing the visual information of user's special actions.

5.2 URC Robot's Smart Action Technology

Smart Action Technology can be divided into the

intellectual covering technology and the intellectual manipulating technology for realizing the physical action of URC (Table II)

TABLE 2 SMART ACTION TECHNOLOGY

Smart Action Technology	The Intellectual Covering Technology	Position-Deciding Technology	The correctly searching technology of robot's position
		Map-Forming Technology	Map-forming technology presented by digital data that grasps the robot's covering environment
		Route-Framing Technology	Route-controlling technology that make the robot plan its course, make it move as the original course
	The Intellectual Manipulating Technology	Environment-Object Cognitive Technology	The cognitive technology that makes the robot recognize the manipulating objects and circumference-environment with using camera, distance sensor
		Manipulation-Control Technology	The technology that manipulates the special object with controlling robot's arms articulation

The Intellectual Covering Technology is divide into 'Position-Deciding Technology' that recognize the present position of robot, ' Map-Forming Technology' that grasps the present operating environment, and 'Route-Framing Technology' that can move safely after generating the moving route for the operation accomplishment. The decision of the moving robot is very easy to human, but very difficult to robot. That is why robot can't possess the intelligence that deals with various uncertainties effectively in real life. In order to develop position-deciding technology applicable to the real environment through URC task, the intensive research is in progress. Map-forming technology is to store the estimated environment information of the periphery in the memory unit from all the sensor information and to use its information if necessary. The most difficult area in map-forming technology is to form the map to the environment sustaining the circulation section.

So far, its level is in the static map-forming level, and it has been developed in the kinetic environment applicable to the technical development. Route Framing method is to cover its course safely, with generating the optimum course toward the final destination, in case of the given map information among the major points, the present and final positions, the optimum factors---safety, time, distance, energy, etc.---when covering. This also generates the optimum course with the real time, securing the safety in the kinetic environment, and makes the research to execute the given task with success by only the least sensor information.

6. TEST CASE DESIGN BASE ON SCENARIO

The Key issue of testing becomes what subset of all possible test cases has the highest probability of detecting the most errors. As one with answers to tin question, we propose a test case design methodology base on base scenario which is made up user requirements.

A well selected test case should also have two other properties:

i. It reduces, by more than a count of one, the number of other test cases that must be developed to achieve some predefined goal of reasonable testing.
ii. It covers a large set of other possible requirements. That is, t tell us something abut the presence or absence of errors over and above this specific scenario.

We define the 5 steps that test-case design by scenario coverage as follows:

i. Identify the workable user requirements in the specification.
ii. Assign a unique number to each user requirements.
iii. Design Test scenario by combining requirement
each requirement should be included by least one scenario.
iv. Until all requirements have been covered by scenario, make new scenario.
v. Design Test cases which can satisfy each scenario.

TABLE 3 USER REQUIREMENT OF HRI COMPONENTS

component	Requirement ID.	Requirement description
Image recognition	SFR.HI.01	Provide a learning tool using "Template Update Algorithm."
	SFR.HI.02	Provide a "back lighted detect function, size of face, as well as GPS verification".
	SFR.HI.03	Provide a "street measuring information" using supersonic sensor.
	SFR.HI.04	Provide a "register and non-register distinction function" through boundary value.
	SFR.HI.05	Provide a "detect function" in 15 degree (up & down, left & right, front & back).
	SNR.HI.PE.01	The target users' recognition rate must reach more than 95%, in three meters of light
	SNR.HI.PE.02	Processing speed should be reached P4, 3G CPU, 1G RAM at 5 frm/sec
	SFR.HS.01	Provide an easy way on-line

Voice Recognition		registration using GUI (speaker registration)
	SFR.HS.02	Provide "independent sentence speaker recognition"
	SFR.HS.03	Provide a "detect function" and "speaker recognition" between family members.
	SFR.HS.04	Provide "strength of signal regularity function" and "choice of microphone" for the long distance transaction purpose.
	SFR.HS.05	Provide "speaker recognition functions" using three or more microphone.
	SFR.HS.06	Provide a "noise filter" and "noise detect function."
	SFR.HS.07	Provide a guidance system after detecting noise.
	SFR.HS.08	Provide a sex and age identification system.
	SFR.HS.09	Provide a adoptable speaker recognition
	SNR.HS.PE.01	"Speaker recognition" and "verification rate" must reach more than 95% when noise level reaches 5~15dB
	SNR.HS.PE.02	While someone using on-line registration, this system must be able to register using two or less sentences.
	SNR.HS.PE03	This system must be able to recognize categories in all four directions within three meters.
Image interaction	SFR.HR.01	Provide a "face detect function" (with in 3m).
	SFR.HR.02	Provide a (head detect function), based on Omega.
	SFR.HR.03	Provide a "follower identification function" using colored clothes.
	SFR.HR.04	Provide "follower identification function" using GPS location system between previous screen and current screen.
	SFR.HR.05	If the Robert looses its follower, then provide a voice message and alarm using TTS.
	SFR.HR.06	If the Robert looses its follower, this system will save pursuit info automatically. After a while, if this system find exact same follower, then he/she will follow the Robert continuously.
	SNR.HR.PE.01	Frontal verification rate must reach more than 95% in one to three meters.
	SNR.HR.PE.02	Personal identification efficiency must reach more than 95% when it does not have occlusion.
	SNR.HR.PE.03	Flank and rear verification rate must reach more than 90%.

7. URC COMPONENT METRIC VS. ISO/IEC 9126 METRIC

External metrics given in ISO/IEC 9126-2 may be used to measure the quality of the software product by measuring the behavior of the system. ISO/IEC 9126-2 consist of characteristics, 30 sub characteristics and 88 metrics. In this Study Table 4 describes a subset metrics of 9126 which are related to quality of URC robot component.

TABLE 4 ISO/IEC 9126 QUALITY METRICS

N/O	Metrics	Sub Characteristics	Characteristics
1	Functional Implementation	Completeness	Functionality
2	Boundary value Coverage	Accuracy	Functionality
3	Functional Implementation Accuracy	Accuracy	Functionality
4	Access Controllability	Security	Functionality
5	Access	Security	Functionality
6	Functional Standard Compliance	Compliance	Functionality
7	Interface Standard Compliance	Compliance	Functionality
8	Fault density	Maturity	Reliability
9	Break Down Avoidance	Fault Tolerance	Reliability
10	Failure Avoidance	Fault Tolerance	Reliability
11	Data Recover ability	Recovery	Reliability
12	Mean Time to Response	Time to Response	Effectiveness
13	Mean Time to Resolve	Time to Respond	Effectiveness
14	Data transfer Rate	Transmission speed	Effectiveness
15	Error Correction	Operability	Usability
16	Error Protection	,,	Usability
17	Message Understand ability	,,	Usability
18	Diagnosis Function	Analyzability	Maintainability

The above table is for testing the robot software components according to the ISO/IEC 9126 standards. This process is still heavily burdened to the robot engineers, since they did not make the software based on the software life cycle method. Most of engineers are working for developing the algorithms and improving the performance. The software technology should be considered for reliability and maintenability with other good attributers. However many engineers have a lack of software skills and only focus on performance.

Considering the background of engineers in robot industry, the evaluation items in the quality metrics were reduced to much smaller number of items. Based on the extracted evaluation items, the robot software components and modules can be tested. The current our research results provides how the testing specifications are made from the user requirements and system specifications.

The next task is to analyze the testing environment, considering the real world and practical usage. We could not provide all kinds of real environments. In order to solve the testing problems, we need to make modeling the various changing environments, where we have to find out what is the most important.

8. CONCLUSION

URC robot offers users necessary services anytime and anywhere, using the high speeded network to the robot, expanding the applicable service, keeping all the functions in the servers, it undertakes functions for the causing problems through the network.

This paper focuses on the testing technology for quality assurance of URC software components. The test case design using scenario is an effective methodology to cover all user requirements. And we are continuously studying to adapt ISO standards for URC component Software testing based an ISO/IEC 9126. This paper suggests the testing process and evaluation items for software testing of robots. This material could be referenced to establish standard software testing of intelligent robots. Nevertheless the test cases for practical evaluation of robots should be studied and developed further more. ETRI is continuously developing the evaluation methods for evaluation of robots.

REFERENCES

[1] H. Kim, "Network based intelligent service robot – URC", ETRI CEO information No.15, 2004

[2] H. Kim, Y. J. Cho, and S. R. Oh, "CAMUS: A Middleware Supporting Context-aware Services for Network-based Robots", IEEE Workshop on Advanced Robotic and its Social Impacts, 2005

[3] ISO/IEC TR 9126, Software engineering – Product quality - Part 4 : Quality in Use Metrics(2003)

[4] Martin Pol, Ruud Teunissen, and E.V. Veenendaal, Software Testing: a guide to the TMap Approach, ISBN 0 201 74571 2, pp.7, 2002

.