Development of Robot Scenario Script Language and Tool for Non-Expert

Donghee Choi [1], Jeayeol Ha [1], Minkyung Jung [1], Woomin Park [2], and Hongseong Park [2]
[1] MZEN, Seoul, Republic of Korea
[2] Department of Electronic and Communication Engineering, Kangwon National University, Republic of Korea
Email: {blesscdh, hjy8099, otgirl20}@gmail.com, {wmpark, hspark}@kangwon.ac.kr

Abstract—**In this paper, we propose a robot scenario tool utilized for assistant robots that are used in a limited area, such as educational institutions, museums and schools. The assistant robot is used to assist operators such as teachers with a given scenario that the robot is operated by time schedule and/or commands, moves the predetermined paths and plays the multimedia contents. In most cases, the operators like teachers are not familiar with computer programming so that we propose a robot scenario script language and tool for non-experts to design their own scenarios. The developers provide example templates and the operator can easily modify the scenario template with GUI to apply their own class scenario and contents. With the proposed robot scenario tool, the user can conveniently define time schedule, the commands by STT (speech to text), the navigation scenario, playing multimedia contents and robot motions. The proposed system including teacher assistant robots and robot scenario tool will be deployed in model schools of Korea next year.**

Index Terms—**robot, script language, scenario script tool**

I. Introduction

As robot technology evolves, the usage field of robot is diversified. In particular, in the field of education / service robot, various methods have been researched and developed to provide dynamic services. Although these robots provide a number of service scenarios to users, the user may not be satisfied with the provided service scenarios. In most case, the customizing cost for on-demand service scenarios is very high to design or modify the service scenario because only expert can utilize scenario tools with expertise such as computer programming and robot science.

The assistant robot performs movement, motion, multimedia playback, interactions with users and so on. In some applications, non-experts cannot design the service scenario due to the complexity of the service. However, in the teacher assistant application, because the space, contents, time and services are limited and predictable, non-experts may be able to design their own scenario with user-friendly tools.

Tools and scripting languages of the robot that has been provided for the development of robot applications have been developing many ways [1]-[8]. These tools provide various ways to control the operation of the robot from the high level of processing of the task of the unit hardware and low-level control. Therefore, the languages and these tools have been developed and optimized for the specific applications. In common with most existing techniques, the author of the scenario configuration of the robot is usually a developer or a technical operator. In other words, the author is a person to understand the characteristics of the robot and may have knowledge of computer programming.

In this paper, we provide a software tool and language dynamically to operate assistant service robot scenario for non-experts. Experts provide to users with example services models and templates. Users (non-expert) are possible to configure, modify and design a scenario easily Drag & Drop a model or a template that experts have provided. The tool includes the function for experts to produce a service model. It is possible to perform all the work in a single application that makes up the service of robots. It provides the functions to communicate between the robot and the person (Inter-action), motion and movement, and multi-media playback. The tool that provides to support (OPRoS [7], ROS [8]), the (C ++, JAVA), and interlocking framework linked language for the operation of the robot. Thus, it includes the ability to be recognized and the service profile used in the framework and functions may be used for the function calls of the programming language, are analyzed by importing interface (C ++, Java) of the language have been. This robot provider makes a basic service model that allows users to configure the scenario of the robot.

Provided tools helps robot providers and end-users.

Robot provider helps to be able to configure the service model easily robotic application system developed.

End-users are able to operate by reconfiguring service scenarios without programming capabilities of the robot.

In South Korea, it is planned to use a robotic system for teaching aids in model schools from 2015. Whereby the robot helps teachers in the class, shows the educational multimedia contents to students and records class history such as the progress of the student. Robot providers are not possible to create on-demand education scenario of all teachers. Therefore, it is desirable to provide teachers the scenario tool to edit the education scenario including multimedia content, navigation and

Manuscript received September 7, 2014; revised December 17, 2014.

doi: 10.12720/joace.3.6.498-502

robot motion that is commonly used through the proposed system for teachers who do not have programming knowledge to perform the scenario creation.

This paper is organized as follows. In Chapter 2, describe the configuration of a service robot system. In Chapter 3, we describe the features and technology of the scripting language of an existing scenario proposed in this paper, and conclusion in Chapter 4.

II. Service Robot System

Fig.1 shows operation of a service robot.

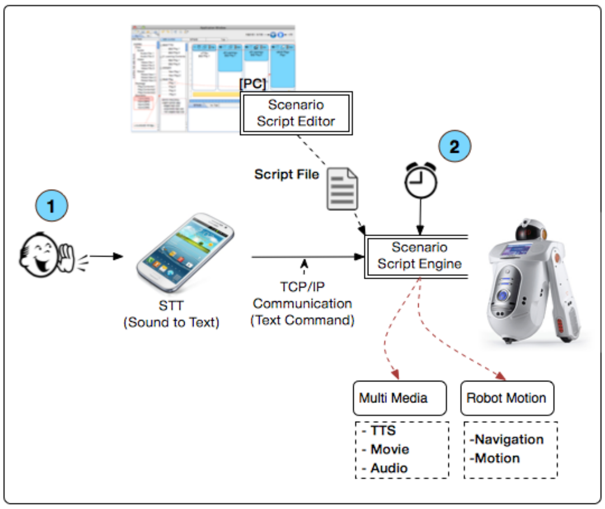

Figure 1. Service robot system

Create a scenario script of the robot, the user sends the robot interpreter scenario script file generated. Then, the user starts the interpreter. Robot will perform the service for a specified time or the user voice command as defined in the scenario created. At this time, the voice command of the user, is done through the application of smart phone or STT(Sound To Text) module of the robot itself. In this system, target reference robot is ED-7280 of ED Corporation. ED-7280 use OPRoS(Open Platform for Robotics Services) Software platform.

III. Robot Scenario Script Tool

Scenario script tool of the robot that was proposed in this paper is shown in Fig. 2.

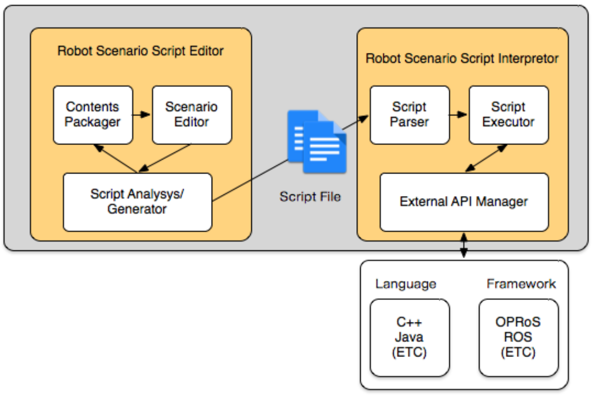

Figure 2. Robot scenario script tool

The tool is composed of two modules. One, create a script, is a module, and the other one is a module to run the script.

A. Scenario Script Editor

The script editor module, to create a script and are performed in the order of operations such as Fig. 3:

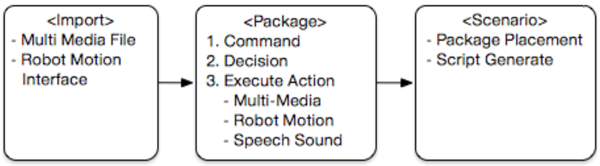

Figure 3. Scenario editor step

- Import: It is a preparatory stage for the scenario creation. To register multi-media contents for player in robot. To register the command for receiving the voice command of the user. To register the software interface to the robot that can perform the motion command.
- Package: This step is of forming a unit of a unit operation. Operation of a service robot that is used in this paper would be to go through request for service operation, service operation, and end step. Here, the service action request is the user's voice command or a time constraint (redefined time, certain period of time has elapsed without an external command from the previous command). Service operation means that the multimedia playback and the robot's motions are executed one or more. If determines whether the repeated, not repeated, end step terminates the service. The package may be constructed a new package into several packages.
- Scenario: This step is to configure the behavior of the robot service unit time.

In step 1, 2, provides a basic configuration robot provider side, users will be using mainly the function of the last step. If a change occurs in the structure of the content at any time, it can be reconstructed by using the format that add or replace only a part of the configuration. (This system does not mean there is another constraint that performs only the last step. However, this system is intended for a non-developer, it is assumed that provided by the logical steps that make up the package.)

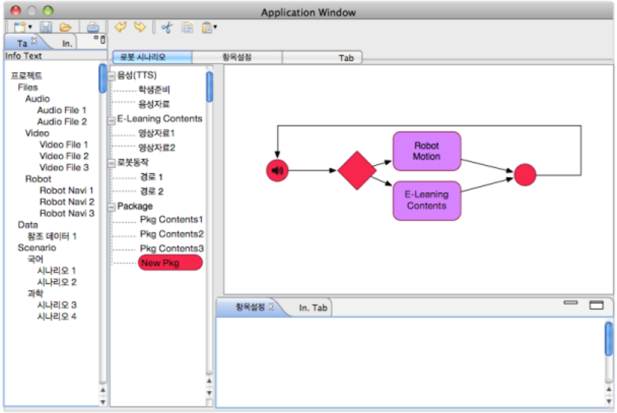

Figure 4. Create package UI

Fig. 4, Fig. 5 are UI Layout of a system for providing a couple of different levels. In step 2, it is an example of a graphical representation of the logical flow of the operation of the package. In step 3, users can configure the scenario along the passage of time packages that was created in the previous step.

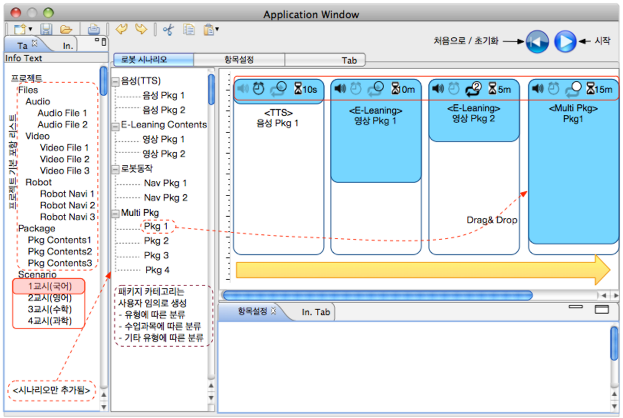

Figure 5. Create scenario UI

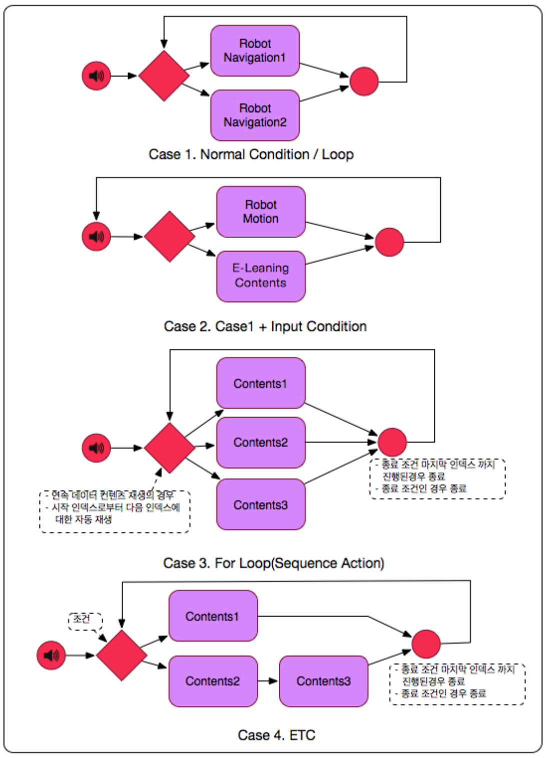
Robot Navigation1
Robot Navigation2
Case 1. Normal Condition / Loop
Robot Motion
E-Leaning Contents
Case 2. Case1 + Input Condition
Contents1
Contents2
Contents3
- 현속 데이터 컨텐츠 재생의 경우
- 시작 인덱스로부터 다음 인덱스에 대한 자동 재생
- 종료 조건 마지막 인덱스 까지 진행된경우 종료
- 종료 조건인 경우 종료
Case 3. For Loop(Sequence Action)
조건
Contents1
Contents2
Contents3
- 종료 조건 마지막 인덱스 까지 진행된경우 종료
- 종료 조건인 경우 종료
Case 4. ETC

Figure 6. Types of package

Fig. 6 shows reference model for package.

The user make flow diagram use graphical icon:

- Start condition(circle) : User voice command
- Selection condition (diamond)
- Loop condition(return line)
- End condition(circle)
- Restart(package or if-condition) or Finish

B. Scenario Script Interpreter

Scenario interpreter module is responsible for execution by analyzing the scenario script file that you created in the scenario editor. Interpreter can be performed with the analysis of the script, and control multimedia playback. It is divided into three (Programming language interface, Robot framework interface) of the external interlocking control services.

The detailed interpreter structure and operation sequence are as shown in the following Fig. 7, Fig. 8.

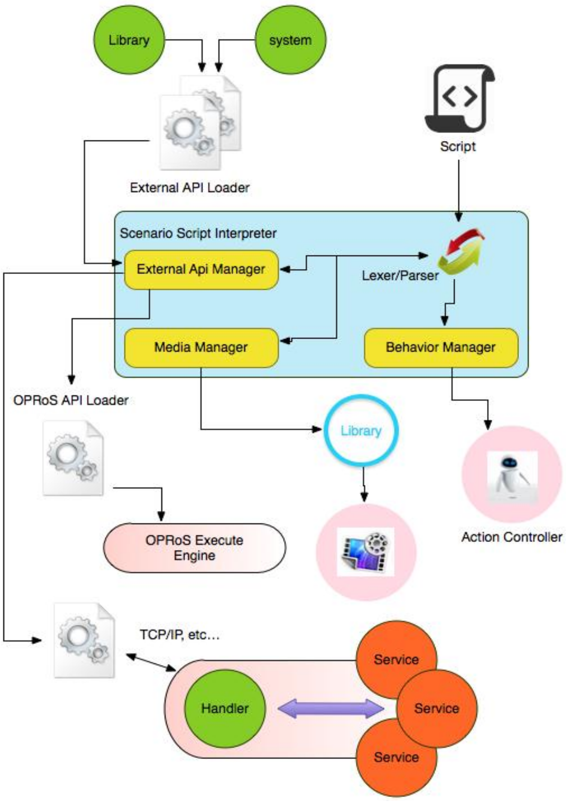

Figure 7. Scenario interpreter module structure

External Api Manager: Load the module reference to the motion of the robot and the media used in editor and check status. The manager performs a function for setting the operation of the robot via the method call of the service framework(OPRoS, ROS) and language library(C, C++, Java) to provide compatibility robot scripting language high level and low level.

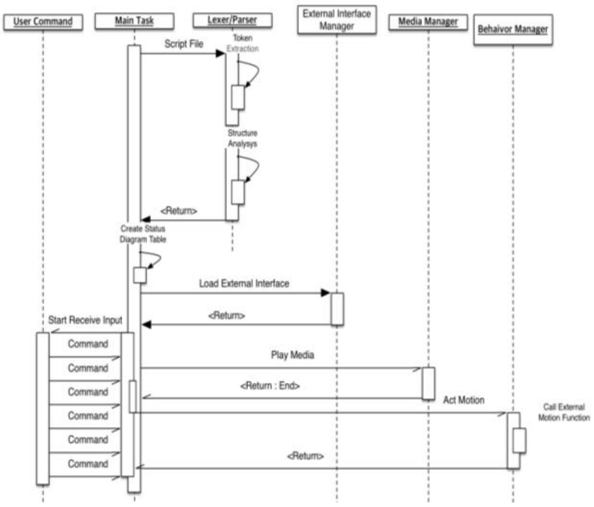

Figure 8. Sequence chart

- Media Manager: Operations to multimedia playback is requested, manage the information of status (execution, waiting, finish). This helps to prevent a duplication operation by the user's mistake, and to efficiently resource management.
- Behavior Manager: Managing the operation of the robot motions. By monitoring the status information of the robot with the progress and motion commands for the robot, and provides a function for the interaction with the user.

C. Definition of Scenario Script Language

Scripting language to be proposed in this paper is applicable to various robots. Language proposed to provide a structure suitable for performing the control on the motion of the robot and multimedia playback. In particular, it must be able to define an interface to access and control the hardware control module (C++, Java), an external framework library that provides hardware access control direct the robot in the motion control of the robot. Also, Programming language, by providing a language representation method of the high level, there is a need to be able to understand quickly and easily by the user. Therefore, it must have a structure suitable to represent the characteristics of these.

Scenario script language is defined in three parts. The first is the definition part of the External API (to define a library that provides a framework and other languages). The second is to create a unit package, and the last is to create the scenario of the package.

1) External API definition

Motion / multimedia playback operation of the robot required to configure the robot scenario requires the function of the external. At this time, it is a portion that defines the external features that require integration.

```
External api = “legacy:///TestAPI”
        void Printf(String message)
        int DriveWheel(double linearVelocity, double
angularVelocity)

End
```

2) Package definition

Define configuration of the unit service for robot. Definition part of each package is composed status structure and status definition. Part of the package definition is configured using the external reference (independent script file).

```
Package
    PkgName
        Init
            //Variable declaration
            //Initial Condition
        End
        Event(event Name)
            //Status Transition
            // Function Call
        End
        Stay
            //Function Call
            //Process Condition, Loop
             Play(user_defined_media_01)
```

```
                //if :if-else
                if(condition)
                    User_Defined_Event_1
                else
                     User_Defined_Event_1
                end

               //for or while loop
                while(condition)
                    Play(user_defined_media_01)
                end

                //End Condition
                if(condition)
                    Package_start()
                else
                    Package_finish()
                end
            End
        End
End
```

Init definition part declares variable and initial condition (user voice command or time). Event definition part is composed of event name and event processing (status transition or function call). Stay definition part is composed of function call, occurs event, logical conditions, and package end process.

3) Scenario definition

It is obtained by sequentially arranging the package. Describe the information about the common attribute (state start, state repeat) of each package.

```
Scenario
    Package : PkgName 1(Package Script File Name)
        Start : //start condition
        End : //end Condition
        Status:
    End
    Package : PkgName 2(Package Script File Name)
        Start : //start condition
        End : //end Condition
       Status:
    End
   Package : PkgName 3(Package Script File Name)
        Start : //start condition
        End : //end Condition
       Status:
    End

End
```

IV. Conclusion

In this paper, we propose a new language and a scenario script tool to edit the education service scenario including multimedia, robot navigation and control of robot motion for non-experts. Robot provider can easily construct a model scenario or a scenario template for the users. End-users can easily modify or make their own scenario with simple operation and service configuration setup. Users can understand quickly and easily how to edit the scenario with the simple and intuitive GUI. The users such as teachers can easily design their own class scenario and share their scenario with other teachers. The proposed system including teacher assistant robots and robot scenario SW tool will be deployed in model schools of Korea next year. It is expected that the proposed system helps Korean students to have interest and focus during class with assistant robots.

ACKNOWLEDGMENT

This work was supported in part by a grant from Ministry of Knowledge Economy in Korea. The project code is 10041789.

REFERENCES

[1] R. Simmons and D. Apfelbaum, “A task description language for robot control,” in *Proc. the Conference on Intelligent Robotics and Systems,* Vancouver, Canada, October 1998, pp. 1931-1937.

[2] At, E., “ESL: A language for supporting robust plan execution in embedded autonomous agents,” in *Proc. the AAAI Fall Symposium on Plan Execution*, 1996, pp. 318-324.

[3] M. Loetzsch, M. Risler, and M. Jungel, “XABSL–A pragmatic approach to behavior engineering,” in *Proc. the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5124-5129.

[4] K. Rockwon, S. Lee, S. Kim, and W. Yoon, “TPL-R: A high-level task programming language for robots,” Unpublished, 2014.

[5] J. C. Baillie, “Urbi: Towards a universal robotic low-level programming language,” in *Proc. IEEE Int, Conf. Intell. Robot Syst.*, Aug. 2005, pp. 820-825.

[6] K. Kushida, Y. Nishimura, H. Dohi, M. Ishizuka, J. Takeuchi, and H. Tsujino, “Humanoid robot presentation through multimodal presentation markup language MPML-HR,” in *Proc. 4th Int'l Conf. on Autonomous Agents and Multi Agent Systems (AAMAS-05), Utrecht*, 2005, pp. 23-29.

[7] C. Datta, C. Jayawardena, I. H. Kuo, B. A. M. MacDonald, “RoboStudio: A visual programming environment for rapid authoring and customization of complex services on a personal service robot,” in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[8] A. R. Khil, K. H. Lee, and S. K. Kim, “Software robot authoring tools for sharing intelligence among users and content providers,” *Multimedia Tools and Applications*, May 2014,

[9] OPRoS. [Online]. Available: http://www.ropros.org

[10] ROS. [Online]. Available: http://www.ros.org

Donghee Choi received the B.S. and M.S in electrical and communication engineering for Kangwon Nation University in 2004 and 2007. He has finished the Ph. D. course degree in department of electrical and communication engineering from Kangwon National University, Chuncheon, South Korea. He is a researcher of research venter of MZEN Co. His research interests include S/W engineering, neural network, multi-agent system, and robot motion.

Jae Yeol Ha received the B.S. and Ph.D. degrees in electrical engineering and computer from Seoul National University in 2003 and 2008, respectively. He is the leader of research venter of MZEN Co. His current research interests include biomechanics, 3D motion capture and analysis, 3D vision, 3D, and Human-Computer-Interface

Min Kyung Jung received the B.S. in information engineering for Myongji University in 2009. She is a researcher of research venter of MZEN Co. Her research interests include image processing, 3D vision, 3D motion capture, S/W engineering

Woomin Park received the B.S. in electrical and communication engineering for Kangwon Nation University in 2012. He is a researcher of Kangwon Nation University. His research interests include analysis of robot software platform, S/W engineering.

Hongseong Park 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 the Department of Electrical and Computer Engineering, Kangwon National University, Korea. Since 2007, he has been the Director of the project called “OPRoS(Open Platform for Robotic Services) developing in KOREA.” His research interests include design and analysis of robot software platform, communication networks and mobile/wireless communication, discrete-event systems, and network-based control systems.