

Networks for a symbiotic Human Life with Robots

Muneyuki Shimokura[†] Shuichi Nakanishi[†] and Tadashi Ohta^{†‡}

[†] Soka University
1-236, Tangi-cho, Hachioji-si, Tokyo, 192-0001 Japan
[‡] ohta@t.soka.ac.jp

Abstract— It is expected that the age when men and robots live together is close at hand. In the age, his/her everyday life will be helped by robots. This article proposes a network software architecture that realizes a space where a man and robots live together. It also proposes a method for describing service programs using a rule based language. For service architecture, a concept of integrated services, which is becoming standard in web services, is adopted. A prototype system for a robot control system was made based on the proposed architecture, and the proposed architecture was confirmed as reasonable.

Keywords-component; Network robot, State transition model, Rule based language, Service orchestration

1. INTRODUCTION

Recently, development of robot technology has been showing remarkable growth, and various robots that can move autonomously and have interfaces to interact with people, have been appearing. The expectation of the birth of a robot, which will help people in all stages of everyday life, is growing. It is anticipated that, as one example of the realizations of ubiquitous networks, the age when humans and robots live together will soon be with us. However, for network service in web services, orchestrating existing services is becoming the main way to develop a new service. This service is called a composite or integrated service.

This article proposes network software architecture that enables a symbiotic human life with robots. Based on the proposed software architecture, a prototype robot control system was made and the proposed architecture was confirmed as reasonable.

2. BASIC IDEA OF SOFTWARE ARCHITECTURE

A network, which enables a symbiosis between human life and robots, is expected as one of the realizations of ubiquitous networks. Robot control is initiated by events such as requirements from a person and information from sensors. Therefore, the processing model of robot control is, in general, considered as an event driven type. So, it can be represented as a state transition model as the basic idea, it can be expressed as a state transition model. Taking this point into consideration, the authors proposed to design network software, which will bring about a symbiotic relationship between man and robots,

based on the state transition model. The authors proposed STAR [3] as software architecture for describing the telephone services programs based on the state transition model using a rule based language. As a service program description language used in the STAR, the authors developed a rule based language, ESTR (Enhanced State Transition Rule), to describe conditions for state transitions and the processes required for the state transitions[3,4]. Taking ease in developing programs into consideration, this paper proposes to describe services programs for robot control using the ESTR.

3. PROGRAM ARCHITECTURE

The ESTR and the program structure based on ESTR are explained. For more detailed explanations, please refer to [3, 4].

3.1 ESTR

Figure 1 shows a form of ESTR.

Pre-condition Event : Post-condition {Action Description}

Fig. 1 Form of ESTR

ESTR consists of a Pre-condition which represents the condition for the rule to be applied, an Event which represents a trigger of the state transition, a Post-condition which represents the condition of a system state after the rule is applied, and an Action description which represents the system processing required for the state transition.

3.2 Program structure

Figure 2 shows the program structure based on the STAR.

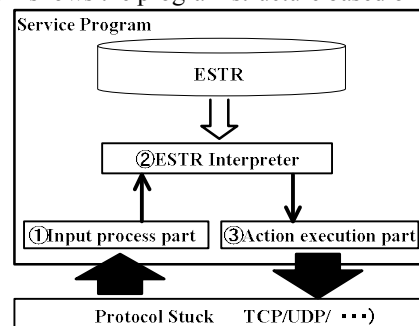


Fig. 2 Service Program block diagram

1) The input processing part converts the received physical signal into a logical signal to be handled by an ESTR interpreter, and sends it to the ESTR interpreter.

2) The ESTR interpreter, after receiving the event, selects a rule to be applied. Rules for applying the ESTR rule are described. When an event occurs, a rule, which has the same event as the event that occurs and whose Pre-condition is included in the system state, is applied. When the rule is selected, the ESTR interpreter sends the Action description of the rule to action execution part. If the process in the action execution part finishes normally, the system state is changed as follows: the Pre-condition of the rule is deleted from the system state and the Post-condition of the rule is added to the system state. Otherwise, the system state is not changed.

3) The action execution part analyzes the action description part sent from the ESTR interpreter and executes the programs corresponding to the action commands.

4. PROBLEMS

Problems of describing a program in ESTR are described.

4.1 Robot Control Method

(1) Robot control for various robots

Capabilities of individual robots and interfaces between robots and a network are different. More concretely, there are differences of sending/receiving signals between a network and robots and the range of capability up to which a robot can move autonomously. Though there are other functional capabilities such as lifting an object, catching an object and so on, these capabilities are out of the scope in this article. Differences of signals mean not only a difference of protocol used by each robot but also differences of the form of the signals and meaning of the signals. For differences in the range of autonomous movement of robots, there are many kinds of robots: some can move to a final destination autonomously, some can not reach the final destination without being given temporal targets. This means that different ways of controlling individual robots are required, even if for the same service. Eventually, it is necessary to describe rules for individual robots, and these results in repetitive description of the rules. Similar problems will occur in adding/modifying services afterwards.

(2) Description of states

Since a robot has many functions, when state transitions of a robot are described in one state transition space, the number of states, which are expressed as a multiplication of each number of states for individual functions, is huge. Consequently, the amount of the description of rules is big. This causes not only a decline in server efficiency but also mistakes in rule descriptions.

4.2 Realization of Integrated Services

In a development of web services, a new idea for developing new services, where the new service is developed by means of orchestration of existing services, is proposed. In

the home network service field, these services are called 'Integrated Services' [2]. To develop programs for robot control services easily and quickly, it is proposed to use the developing model of Integrated Services. To develop services based on the developing model of Integrated Services, interfaces of individual services should be defined so that individual services can be called by any service and can be returned correctly. As concrete interfaces, events for initiating individual services, called 'initiation event', and events for notifying end of service execution, called 'end event', should be considered. There are some cases for initiating individual services in the Integrated Services, such as: the case where one service initiates a number of services, the case where some services initiate one service, and the case where one service initiates one service more than once.

In the case where one service initiates multiple services, on receiving an end event, the service should identify from which service it receives the event.

In the case where multiple services initiate one service, when the initiated service sends an end event, the initiated service should know to which service the end event should be sent

In addition, there is a case where after a service (for example Service A) initiates another service (for example Service W1), A initiates W1 before W1 completes. In this case, when W1 ends one of the requirements, it is necessary for A and W1 to know which requirement to W1 has ended.

To summarize what was discussed above; when the initiation event is sent, it is necessary for the information to identify from which service the initiation event is sent, to be added so that the initiated service can send the end event to the service which sent the initiation event. And then, it is necessary for the service, on receiving the end event, to identify which process should be restarted.

5. SOLUTIONS

The solutions for the problems described in section 4 are discussed.

5.1 Robot Control Method

(1) Robot control for various robots

As for differences of sending/receiving signals which differ depending on individual robots, the differences are resolved in the Input process part and the Action execution part shown in Figure 3. Concretely, for differences of signals sent to the robot control server, protocol names and signal data formats used by individual robots have been registered in the robot registration table of the Input process part beforehand. On receiving an input signal, the Input process part specified the robot, based on the originating IP address using a robot registration table. The Input process part identifies the received signal based on the specified robot and converts it to a logical event which the ESTR interpreter can understand (Figure 3).

Differences of signals sent to the robot from the robot control server are resolved in the Action execution part which

sends control commands to robots. Concretely, protocols and signal formats corresponding to individual robots have been registered in the robot registration table of the Action execution part beforehand. When an action command is executed, the Action execution part converts logical signals described in the action command to the physical signals corresponding to the designated robot by referring to the IP address and signal format for the robot registered in the robot registration table, and sends the physical signal using a protocol corresponding to the robot (Figure 3).

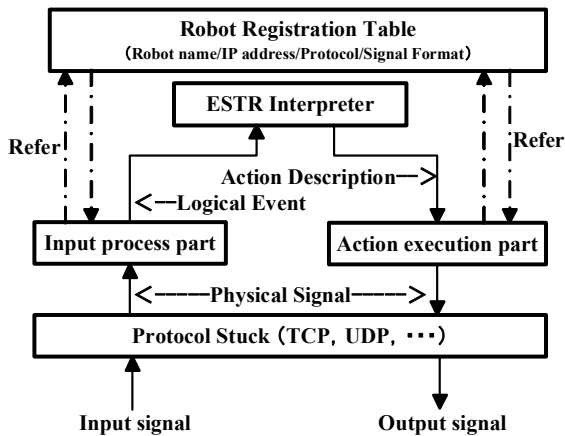


Fig. 3 Conversion from a logical signal to a physical signal

A method for resolving the differences of capability in autonomous movement is proposed. The essential function for controlling robot movement is to move the robot from the present position to the next destination. So, the way of controlling robot movement can be described as the common rules (Figure 4).

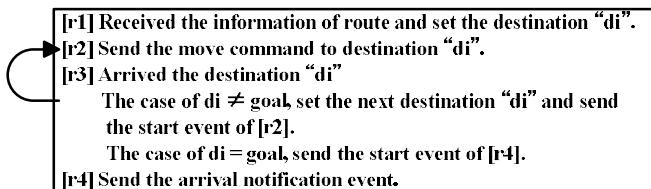


Fig. 4 A set of rules for moving the robot

The control of moving robots consists of the following three common processes:

- Selection of the destination
- Movement of robots to the destination
- Judgment of the final destination

The robot control server can lead the robot to the final destination by repeating the three common processes described above as many times as needed.

(2) Description of states

To prevent an increase in the number of states of robots, the state transitions of robots are not expressed as one state transition space but they are divided into several state transition spaces. Concretely, multiple functions of robot are

taken into consideration. In general, there are functions that can work independently from other functions. All functions are classified into several categories. Functions classified into different categories can work independently. The state transition space of a robot is divided into the state transition spaces that correspond to the categories.

5.2 Realization of Integrated Services

The problem in identifying the service to be initiated is solved by defining the initiation event uniquely for each service. Similarly, the problem in identifying the service, which sends the end event, is solved by defining the end event uniquely for each service. Suppose, the initiation event and the end event of W_i , shown in Figure 4, are defined as "staevwi" and "endevwi", respectively. To identify which service initiates, the subject in the service is taken into consideration. For the control system, in many cases, the object to be controlled is the subject of the service. For the services which control the robots, in most cases, the robots are subjects of the services. Let y be the subject of the service which initiates and let x be the subject of the initiated service. In this case, the service which initiates requests the initiated service to control x for some purpose. When the initiated service completes the requirement, it notifies y that the control to x was completed. On receiving this notification, the service which initiates the initiated service restarts to control y .

Here, let's assume the following: the service, which receives a request for controlling x from the service whose subject is y , does not accept any new requirements to control x from services whose subject is y before the control to x is completed. Thus, the service to which the end event is sent and the process to be restarted in the service can be identified by the combination of x and y .

Namely, by defining the initiation event and end event as $staevwi(x,y)$ and $endevwi(y,x)$, respectively, it can be identified which service initiates which service. The process to be restarted can also be identified (Figure 5).

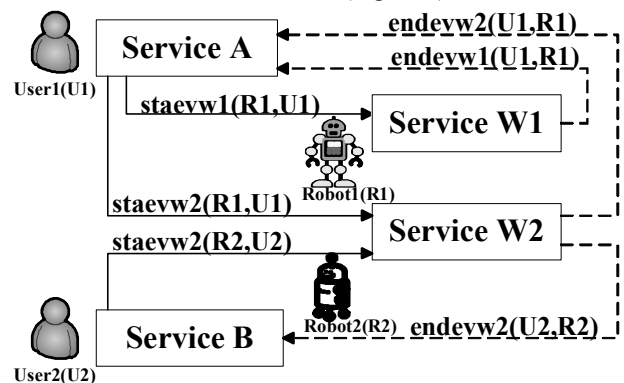


Fig. 5 Example of events in the integrated services

6. PROTOTYPE SYSTEM AND EVALUATIONS

To evaluate the solutions proposed in section 5, a prototype system which controls AIBO was developed.

6.1 Prototype system

As a prototype service, Visitor Guide Service was developed. In Visitor Guide Service, when a visitor comes in the entrance hall, a robot goes to the entrance hall to greet the visitor and guides the visitor in accordance with the visitor's requirements. First, as functional elements to be commonly used as service components, Robot Control Function, Robot Guide Function in a section, and Conversation Function were developed. Next, using functional elements developed beforehand as described previously, as a single service, Robot Guide Service, Conversation Service, and Path Selection Service were developed. Then, as integrated services, using Robot Guide Service, Conversation Service, and Path Selection Service, and functional elements, Visitor Reception Service and Navi Service were developed. Finally, by orchestrating Visitor Reception Service and Navi Service, and using functional elements, Visitor Guide Service was developed (Figure 6).

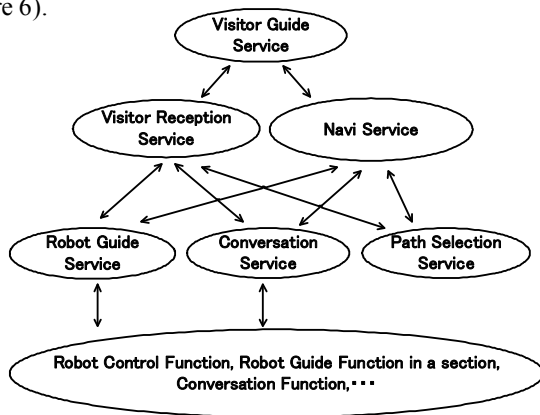


Fig. 6 Service block diagram

The system structure is shown in Figure 7. AIBO has the capabilities to search for a pink ball and to reach the ball autonomously. A service program was developed using these capabilities.

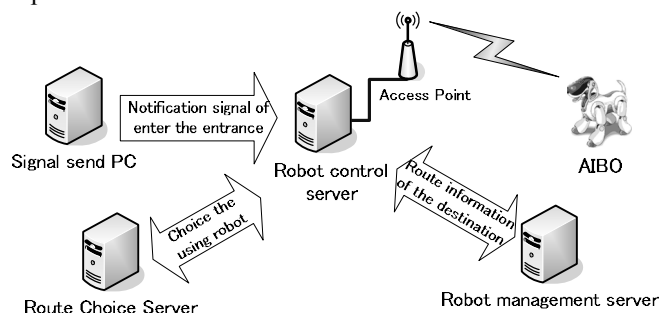


Fig. 7 System structure

6.2 Evaluation

(1) Robot control for various robots

By modifying the program of AIBO, two AIBOs with different capabilities in moving autonomously were made. It was confirmed that the two AIBOs can be controlled by the same rule set.

(2) Description of states

The states of AIBO are classified into five categories: states related to the robot control function (target search and target approach), states related to the Robot Guide function, states related to the Conversation function, states related to the Robot Guide Service, and states related to the Conversation Service. The number of states for individual categories is 5, 3, 2, 2, and 4. If these states are described in one state transition space, the number of states is 120. By dividing the state transition space into five categories, the number of system states can be reduced to 16, and the effectiveness of the proposed solutions was confirmed.

(3) Realization of Integrated Services

Based on the proposed solutions, three functional elements and three single services were developed. Then, two Integrated services were developed. Finally, by orchestrating all of them, an Integrated service was developed. It was confirmed that the Integrated service worked correctly when multiple users operated multiple robots simultaneously.

7. CONCLUSION

Based on the state transition model, the network software architecture for realizing a symbiotic human life with robots was proposed.

For future work, from now on, the system will be improved so that, in the case where the same subject initiates successively the same service for the same subject of the service, the end event can be identified correctly. In addition, an experiment, actually using different robots, will be done to confirm the proposed methods as reasonable.

ACKNOWLEDGEMENT

This research was partially supported by "Collaboration with Local Communities" Project for Private Universities on "Development of Ubiquitous-Monitoring-Network Based on Distributed Sensor Nodes using Local Positioning / Optical Sensory Nerves and their Industrial Applications" : matching fund subsidy from MEXT(The Ministry of Education, Culture, Sports, Science and Technology of Japan), 2006-2010.

REFERENCES

- [1] Woojin Chung Gunhee Kim Munsang Kim Chongwon Lee "Integrated navigation system for indoor service robots in large-scale environments", Proc. of ICRA, 5099- 5104 Vol.5. 2004.
- [2] Masahide Nakamura, Hiroshi Igaki, Ken-ichi Matsumoto, "Feature Interactions in Integrated Services of Networked Home Appliances -An Object-Oriented Approach-", Proc. of Int'l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI'05), June 2005.
- [3] T. Morinaga, G. Ogoe, and T. Ohta, " Active Networks for VoIP GW using Declarative Language", Proc. of APCC2001, pp.89-92, Sep. 2001.
- [4] T.Yoneda and T.Ohta,"The Declarative Language STR (State Transition Rule)," Proc. of FIREWorks, pp.197-211, Jun 2000. Springer.