9

Speech Communication with Humanoids: How People React and How We Can Build the System

Yosuke Matsusaka
National Institute of Advanced Industrial Science and Technology (AIST)
Japan

1. Introduction

Robots are expected to help increase the quality of our life. Robots are already widely used in industry to liberate humans from repetitive labour. In recent years, entertainment is getting more momentum as an application in which robots can be used to increase peoples quality of life (Moon, 2001)(Wada et al., 2002).

We have been developing the robot TAIZO as a demonstrator of human health exercises. TAIZO encourages the human audience to engage in the health exercise by demonstrating (Matsusaka et al, 2009). When demonstrating, TAIZO, the robot and the human demonstrator will stand in front of the human audience and demonstrate together. For this to work, the human demonstrator has to control the robot while they themselves are demonstrating the exercise to the audience.

A quick and easy method for controlling the robot is required. Furthermore, in human-robot collaborative demonstration, the method of communication used between the human and robot can be used to affect the audience.

In this chapter, we will introduce the robot TAIZO, and it's various functions. We also evaluated the effect of using voice commands compared to keypad input during the robot-human collaborative demonstration. In Section 2 we explain the social background behind the development of TAIZO. In Section 2.5 we will discuss about effects of using voice commands compared to key input in human-robot collaborative demonstration. In section 2.6 we present an overview of the system used in TAIZO. In Section 2.7 is the evaluation and discussion of the results from experiments in which weãĂĂmeasured the effect of using voice commands through simulated demonstrations. Finally, in Section 2.10 and Section 2.11 we will discuss about the benefits and problems of using humanoid robot to this application.

In latter part of the chapter, we will discuss how to develop the communication function for the humanoid robot.

Recently, "behavior-based" scripting method is applied in many practical robotic systems. The application presented by Brooks (1991) used hieratical structure model, the recent applications (Kaelbling, 1991) (Yartsev et al, 2005) uses state transition model (finite state automata) to model the situation of the system. The developer incrementally develop the script by adding each behaviors which fits to each small situations. Diverse situation understanding ability can be realized as a result of long-term incremental development.

The behavior-based scripting method can also be applied to communication robots by incorporating speech input with the situation model. Application of the behavior-based

scripting method to the communication robot is first presented by Kanda et al (2002) in 2002. In their work, they not only proposed an incremental development framework, but also implemented an on-line development environment which can realize automated control of the robot. They have confirmed through 25 days field study that with help of the development environment, the conversation ability of the robot was incremented on-line and succeed to decrease operation time of the human operator (Kanda et al, 2009).
However, in the existing behavior-based scripting methods for communication robot, there is an inefficiency in terms of reusing the script to develop different types of robots (this problem is described in Section 3.2.2). This chapter, we present the extension-by-unification method in order to push forward the behavior-based scripting approach to develop communication robots.
In Section 3.1, a component based architecture we have developed that can develop a robotic dialog system only by connecting a components will be shown.
In Section 3.2, a formal discussion of an incremental development methods for the state-transition model is presented. Here, we introduce the formalization of the proposed incremental development method and clarify its characteristics by comparing it to the previous method.

2. TAIZO robot and robot-human health exercise demonstration

2.1 Background: Aging society in Japan

The aging society is becoming a serious problem in Japan. Like many developed countries, the decrease in birth rate and advance in life expectancy is proceeding very steeply. Due to the post war baby boom, the population of the elderly (over 65 year olds), has exceeded 20 percent of the whole population.
In the past, social welfare service has been focused on giving good medical care to the elderly. But recently, attention is shifting towards minimizing the needs of medical care itself.
Minimization of medical care not only has financial benefits, but is also important for increasing the individuals quality of life. By keeping good health and not needing medical care, an elderly person can continue to make their own decisions. This is rarely possible if they are hospitalized. Health exercises are considered to be an effective way to keep their health and minimize medication.
There is a lot of activity going on for spreading the health exercises. In the Ibaraki prefecture, the local government and a local university has co-developed a program for teaching health exercises. One characteristic of the system developed in the Ibaraki prefecture is that the trainer of the exercise is also an elderly person. The prefecture gives certificates to elderly people who have mastered the specified teaching program. This in turn qualifies them to participate as volunteers to teach the exercise to other elderly people. Due to this elderly-to-elderly teaching system, the number of health exercise demonstrators has increased to over 2300.
TAIZO is developed to assist spreading the health exercise.

2.2 Humanoid robot as an exercise demonstration medium

We use a humanoid robot as a medium to demonstrate the exercise, because the following points make it an effective demonstrator.
Similarity of the shape of body: Because a humanoid robot is designed to imitate the shape of a human, a person can easily observe and imitate the demonstrative body motion expressed by the robot.

Attraction: Meeting face to face with a humanoid robot is not yet a common occurrence. It is easy to spark the curiosity of someone who is not familiar with humanoid robots and grab their attention with an artificial being which moves and talks.
Embodiment in 3D space: The robot has a real body which has an actual volume in 3D space. Compared to the virtual agents which only appear in 2D display, it can be observed from very wide view points (Matsusaka, 2008) (Ganeshan, 2006). From this characteristic, even the user at the back of the robot can observe the robots motions. A user to the side can also observe other users interactions with the robot.
Similarity of the body shape assists precise communication of the body movement to the trainee and useful to enhance the effectiveness of the exercise. Attraction also becomes a good incentive for the trainee to engage in the exercise. Embodiment in 3D space assists the social communication between the robot and the trainees. A trainee can watch other trainees while training. By looking at other trainees, they can see how well they communicate with the robot and how eager they engage in the exercise. In most of our demonstration experiments, this inter-audience peer-to-peer effect gives positive feedback to enhance the individuals eagerness to engage in the exercise (explained and discussed in section 2.9).

2.3 Demonstration setup and scenario

Health exercise is intended to strengthen the body of elderly people. It consists of several kind of exercise that involve stretching and muscle training. Health exercise is recently increasing in importance and is gaining attention as an effective way to reduce the number of elderly people who become bedridden and need nursing care and increase their quality of life. Despite these good points, there are still some difficulty in applying this illness prevention activity. One of the biggest problems is the difficulty to encourage people to engage in these health exercise to begin with.
People are not wary of their health while they are healthy. They notice after they realize they have a serious illness. Although, we have statistics on the percentage of people who become seriously ill, it is still difficult to estimate how healthy we are ourselves. Moreover, it is much more difficult to understand the effect of health exercise, since we cannot compare what would have happened if we didn't (or did) engage in the exercise.
Because of above, we usually require a special incentive to engage in the health exercise.
TAIZO (Figure 1) is designed to help demonstrate health exercise. It will stand to the side of the demonstrator and assist the trainer in demonstrating. Demonstration is usually done in front of 5 to 15 trainees. TAIZO is used to demonstrate at events with 40 to 80 trainees. The robot is used as an eye-catcher to capture the attention of people who don't know the exercise, but could be a potential regular trainee. By using the robot as a demonstrator, human demonstrator can get more interest from a larger variety of people compared to a demonstration done by humans alone. This leads to more people having a chance to engage in health exercises.

2.4 Role of the human demonstrator and the robot

Both human demonstrator and the robot stand in front of the audience (Figure 2). Human demonstrator leads the training program and the robot follows. Both human and the robot show the demonstration to the audience.
To follow the human demonstrators lead, the robot has to accept commands given by the human demonstrator. In addition, although the human demonstrator takes the lead in most situations, the robot has to collaborative with demonstration activity in order to make it more

Fig. 1. TAIZO robot

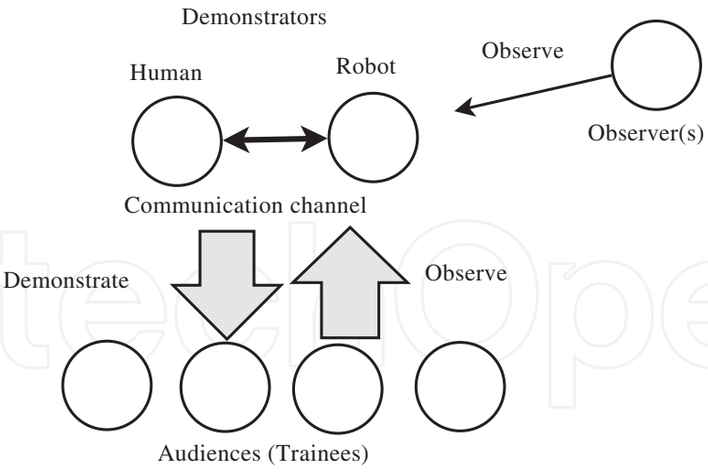


Fig. 2. Demonstrative setup

attractive. The collaboration activity itself could sometimes be an attention catcher to increase attendance of the demonstration. This chapter focuses on this communication effect in the human-robot collaborative demonstration.

2.5 Difference of communication channels in robot-human demonstration

2.5.1 In-demonstration conversation

When demonstrating the exercise there is a dialog between the trainer and the supporting robot. We call this dialog "in-demonstration conversation".

There exists research which handles this in-demonstration conversation. Katagiri et al (2001) has realized and evaluated the effect of demonstration through virtual agents which uses in-demonstration conversation. In Japan, this dialog has been developed into a two-man stand-up comedy called manzai. There several pairs of robots which have been used for manzai to entertain a human audience (e.g. Hayashi et al (2008)).

However, in most of previous research, the demonstration setup consists of only two robots. If one of the members becomes human, there will be a communication issues between the human and robot. In this chapter, we introduce our health exercise robot which has handled the communication issue specific to a human-robot setup.

2.5.2 Speech or keyboard : Discussion from both sides

In human-robot collaborative demonstration, what is the most appropriate method to give commands to the robot? Here, we compare two input methods, key input and vocal input.

When we compare the two input methods from accuracy, key input is more precise than the speech input.

When we compare two the input methods from their characteristics, key input uses a private channel, while vocal commands are public. In other words, vocal commands can be heard from the audience, but key input cannot be heard or observed by most of the audience.

From the demonstrator's side of view, a more precise input channel may be preferable, because they don't want to make errors. But when we consider the audience's point of view, a more transparent communication may be more preferable. From this point of view, vocal commands may be preferred, because it will allow the audience to observe the flow of the dialog between the human demonstrator and the robot. Interaction using an audio medium is publicly observable, and from a users standpoint, could overcame keypad input despite it's inaccuracy.

In this chapter, we will not only evaluate the effectiveness of each input method from a demonstrator's point of view, but will also focus on the demonstrative effect to the audience. In Section 2.7 evaluation design is described.

2.6 Architecture of TAIZO robot

2.6.1 Speech input system

Figure 3 shows the overall architecture of TAIZO robot.

Speech input system consists of a speaker-phone device (Figure 4) to capture and emit sound. The speech recognition system Julius (Kawahara et al, 2000) inputs the captured sound and outputs the recognized phrase. The recognized phrase is matched with a phrase database in the phrase matcher. The phrase database contains a set of phrases written in a form of script which associates phrases and commands. Details of the speech I/O system will be shown in Section 3.1.

2.6.2 Key input system

Key input system consists of a keypad device (Figure 4). The human demonstrator types a number using this keypad.

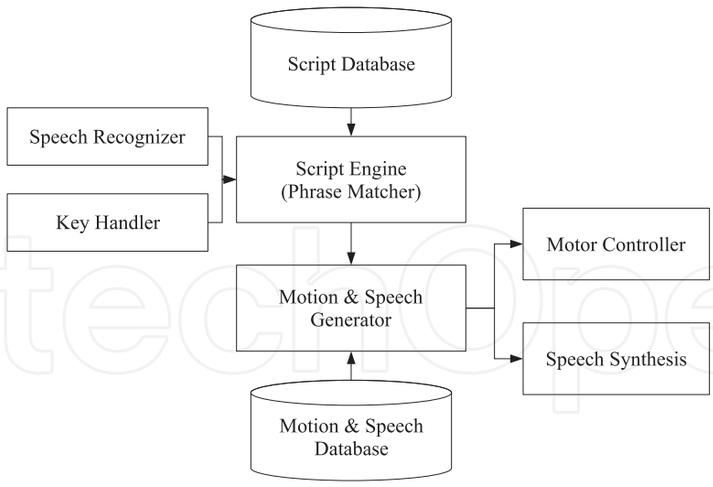


Fig. 3. System architecture

Fig. 4. Keypad (left) and speaker-phone (right)

The keypad is used for both key input mode and speech input mode. In speech input mode, the keypad is used to control the volume of the microphone device. Input to the microphone device switches on when the key is pressed, and turns off when the key is released (push-to-talk).

2.6.3 Motion and speech database

The motion database contains sequences of values specified to each joint of the robot. These sequences are designed to express the exercise motion of the robot.

In this chapter we have designed 17 motions (see Figure 5). When designing the motion, we first create an abstract design of the motion using a robot simulator (Hirukawa et al, 2003). Final adjustment is done by playing back the sequence on the real robot.

Fig. 5. Motions designed for health exercise.

We play back prerecorded speech for the robot. Patterns of prerecorded speech consists of exercise related phrases (e.g. "Raise your arm behind your head", "Twist your waist"), question answer type phrases ("Yes", "Okay", "No") and greetings ("Hello", "Good Bye"). Exercise related phrases are recorded by the script developer, each time a new exercise is designed. The script developer can also write a script for interaction by using prerecorded QA type phrases.

Type	Text (followed by "Do you ...")	When
Expectation	think the demonstration by the robot effective?	Before
Enjoyableness	think the demonstration was enjoyable?	Each
Easiness	think using robot is easy?	Each
Fulfillness	think the demonstration was fulfilling?	Each
Preference	prefer speech input or key input?	After
Effectiveness	think the demonstration by the robot effective?	After
Willingness	want to use the robot in real demonstration?	After

Table 1. Items in question sheet (each question allows free commenting).

2.6.4 Motion and speech generation system

During motion generation, motion data in the database is transmitted to the motor controller. Motion data is transmitted sequentially at a specific rate using the internal clock of the robot. Speech recordings are synchronized in parallel with the motion at specified timings.

2.6.5 Script engine and script database

The scenario database contains scripts which define speech phrases and key numbers of the keypad commands. It also includes information which associates the commands to motion data.

The script engine is based on a state transition model. For this specific experiment, we use a flat (one-state) structure model with 17 commands. Details will be shown in Section 3.2.

2.7 Evaluation of command input methods

2.7.1 Experiment condition

We have designed 4 experiments controlled by two conditions. Condition 1 is the input which be either keypad input or vocal command. Condition 2 is defined by the subjects role which is demonstrator or audience. The conditions regarding the role of the demonstrator switches when the same subject experiences the same demonstration as a demonstrator or as a passive participant. Each subject is asked to attend all of these 4 (2×2) experiments.

Experiment subjects are 60-80 year old, and consists of 1 male and 5 females. All subjects have experience in the health exercise training program and are certificatified to instruct. Although they are knowledgeable in the health exercise, when participating as passive audience, we asked them to answer the questions (described in next section) as if they are a novice trainee.

2.7.2 Experiment sequence

The six subjects were divided into two groups of three. There are six sessions altogether for each group. The first three sessions used the keypad and the latter three sessions used vocal commands. In each of the three sessions, three subjects in turn take the role of demonstrator.

Each subject is asked to fill in a questionnaire sheet after each session. Table 1 shows the questions used in this experiment. Questions consist of asking the ease of use, whether the experience was enjoyable and whether the demonstration was fulfilling.

Before and after the experiment, the subject is asked to fill the question sheet. In the questionnaire handed to the subject before the experiment, expectation for the robot is asked. In the post experiment questionnaire, the subjects preference to keypad or vocal commands and willingness to continue using the robot in a real demonstration is asked.

2.8 Result

Figure 6 shows the error rate of the key input and the speech input.

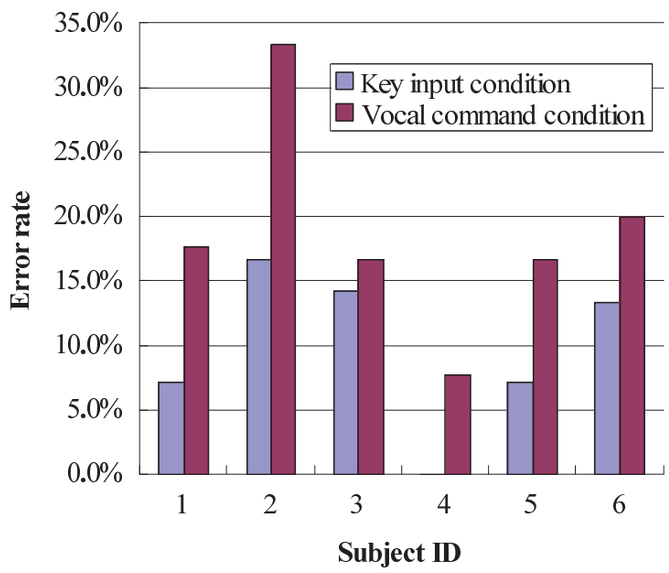


Fig. 6. Error rate of each input methods.

Although the key input is a precise input method, there are some errors due to mistyping. Mistyping can be classified into two types. One is typing error, which happens when the demonstrator is in a hurry to type the keypad in the middle of the demonstration. The other is memory error. Memory error happens because the keypad only accepts numeric input and the demonstrator has to remember the mapping between the number and the training pattern. They often forget the mapping and type the wrong number. Because of these errors, actual precision of the key input is not as high as we expected.

Most of the errors in vocal commands happen due to speech recognition errors. Some demonstrators had problems in pronunciation and other demonstrators spoke superfluous words to enhance the demonstration. In the case of vocal input, there was less memory error, because the demonstrator only has to pronounce the name of training pattern and there is no need to remember the mapping to the numbers.

Figure 7 shows the impression of the demonstration asked after each session. Demonstrator feels using the vocal command is easier than the key input. This can be understood from the memory error as we discussed above. When vocal commands were used, the audience both enjoyed the demonstration and found it as fulfilling as the keypad input demonstration, despite happenings due to inaccuracy. This could be one of the effects of the observability of speech in a human-robot collaborative demonstration.

Figure 8 shows the impression of the robot assisted demonstration before and after the demonstration. Almost all the subjects have answered that they are willing to use the robot as an instructor again.

2.9 Feasibility test in the real demonstration

We have already started applying this robot in real demonstration events. Figure 10 shows photos from the "Nenrin-pic" event. Nenrin-pic event is one of the event hosted by Ibaraki prefecture intended to encourage sports for elderly people.

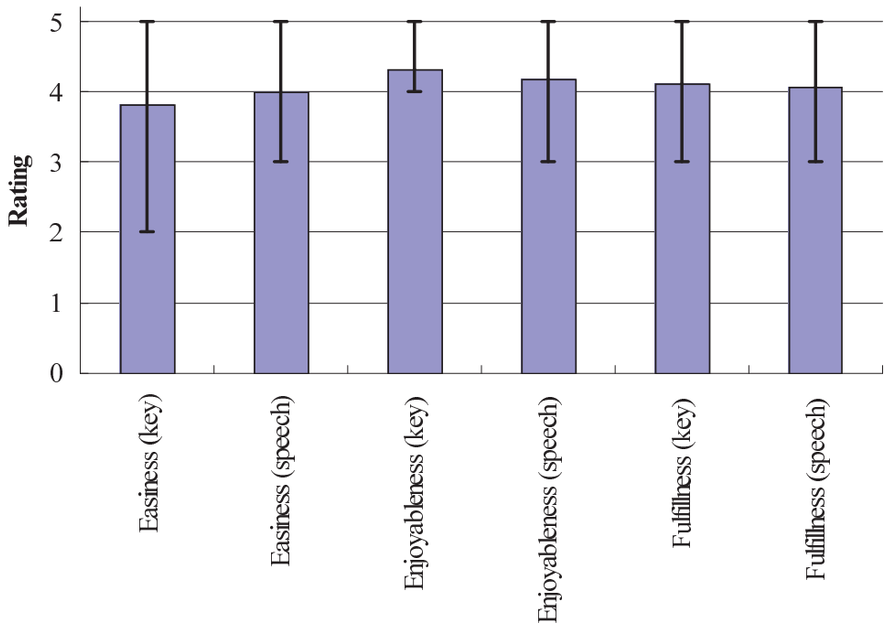


Fig. 7. Impression of the demonstration asked after each session. The error bar indicates maximum and minimum rating of the subjects. Easiness is asked to the demonstrator. Enjoyableness and fulfillness are asked to the audience.

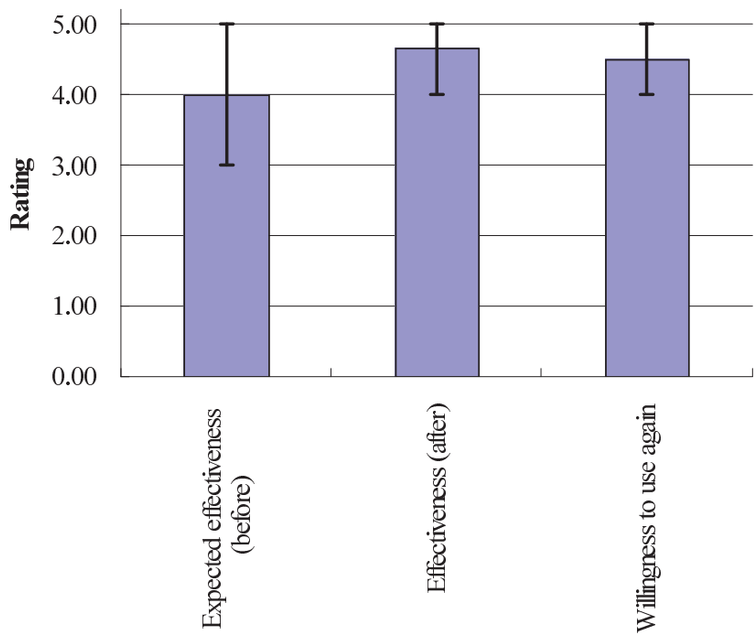


Fig. 8. Impression of the robot before and after the demonstration. The error bar indicates maximum and minimum rating of 6 subjects.

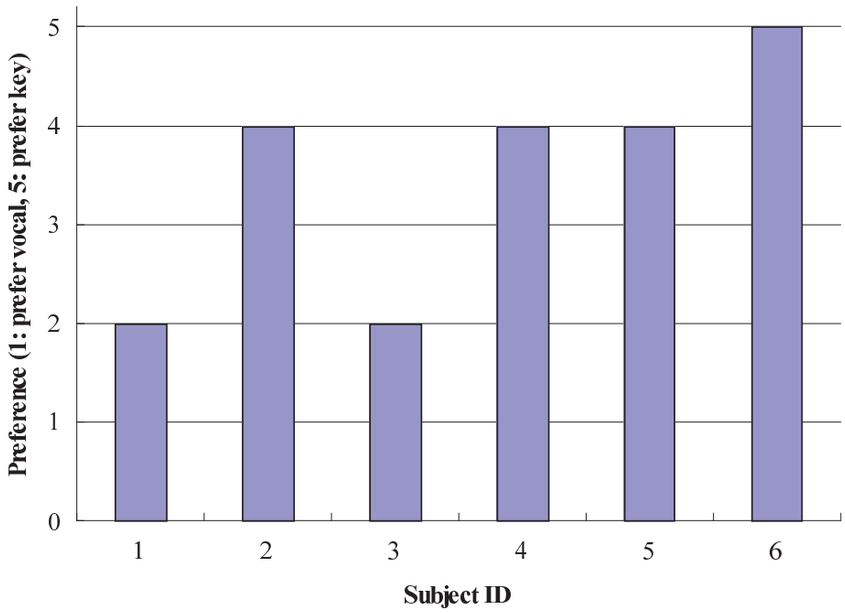


Fig. 9. Preference of using speech input or key input asked to each subject.

Fig. 10. Photos from a real demonstration at nenrin-pic event.

During the 3 day event, we have demonstrated 10 times a day using TAIZO robot. More than 600 peoples has joined the training experience. As we can see from the photos, almost all the audiences were eagerly followed the demonstration given by the human-robot demonstrators. One of the unexpected effects of using TAIZO was that we were able to catch the attention of a wide variety of ages. TAIZO was intended to catch attention of the elderly, but during the nenrin-pic event, many young adults and their children accompanying their parents or grand-parents were drawn to the demonstration. The attraction of the TAIZO robot was strong enough to catch also those accompanying persons. It seems to have a good effect for the elderly, because they can enjoy their exercise by participating together with their families.

2.10 Summary

In Section 2.7 we have tried to evaluate effect of using key and speech inputs especially focused to human-robot collaborative demonstration setup.

As we discussed in Section 2.5, vocal communication increases the transparency of the human-robot communication to the audience. We could see from the experiment results that the enjoyment and the fulfillment of the demonstration from audience perspective was not let down by the imprecision in speech recognition.

One of the unexpected effect of using speech is, because it is very intuitive, it decreases the burden for remembering the commands. Error rate of key based commands is unexpectedly high, despite it's preciseness. This may also support the use of speech input.

Despite supportive evidence for speech input, about half of the demonstrators answered that they prefer using key input. In the question sheet, subject can leave comments. Subjects who are supportive to the speech input commented that they actually enjoyed reacting to mistakes made by speech recognition. This comment can be explained that by increasing the transparency of communication channel between human-robot demonstrators, the subject can observe what is happening in the situation (human demonstrator said the right thing, robot mistakes) and feel the mistaken response of the robot funny. On the other hand, subjects who are supportive to the key input commented that they want to demonstrate the exercise in a more precise manner.

We are currently preparing to do an evaluation which is more focused on measuring these effects and searching for a way to realize an appropriate interface for both people who prefers enjoyment or precision.

2.11 Left problems

As we have looked and discussed in this section, humanoid robot has different character than the other artifacts. *It has human shape* which can attract human to join in the activity. This character also gives some effect to *gain expectation to use natural communication method (voice)* as the human do. *It has physical body and exist in same world* which can give effect also to the observers. These characters are especially useful for applications such as exercise demonstration.

However, this character sometimes gives negative effect to the usefulness. Because the human gain too much expectation to the robot to use natural communication method, human tends to use colloquial expression towered the robot, which is difficult for the robot to understand. In Section 2.7, we have seen the command acceptance rate using voice recognition evaluated by elderly users. In this experiment the command acceptance rate is low, not because voice recognition rate is low, but mostly because conversation patterns programmed to the dialog manager was not enough to understand the all varieties of colloquial expressions given by the users.

Because the robot has physical body and exist in physical world, the voice recognition system of the robot have to work under noisy condition of the real environment.

Although for ideal benefits of using humanoid robots, above practical problems are need to be solved beforehand to enhance the usefulness of the robots.

We are not only developing the applications for humanoid robots, but also developing a support tools for assist development of the communication functions for humanoid robots. From the next section, we will introduce our development tools.

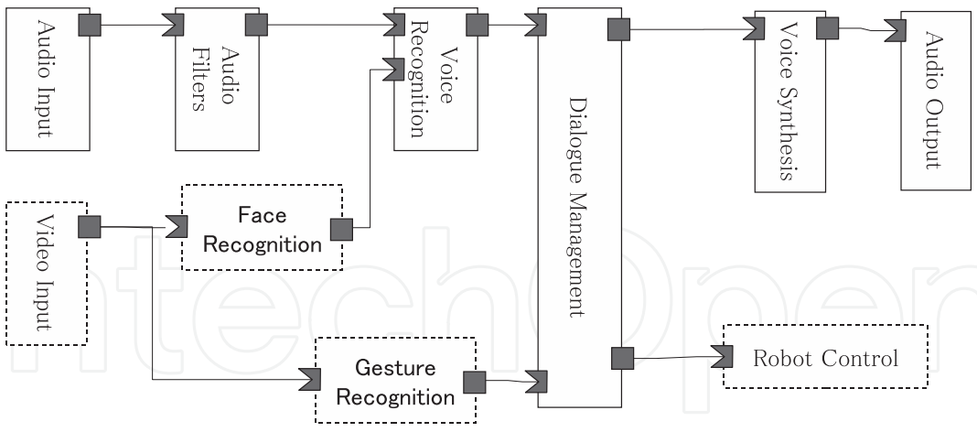


Fig. 11. Architecture of the OpenHRI software suite.

3. Reduce the difficulties of building the communication system

In this section, we introduce our efforts to reduce development difficulties.
We are currently taking two approaches to reduce the development difficulties. The one is component based system design and the other is incremental script development.

3.1 Component based system design

At present, we are developing a set of software called Open Source Software Suite for Human Robot Interaction (OpenHRI). Using OpenHRI, we aim to solve the above problems and enable the development of communication functions for robots. For this purpose, we employ the following approach.
Introduce a uniform component model: We construct our set of software on RT-Middleware, an object management group (OMG)-compliant robot technology middleware specification (Ando et al, 2005). The RT-Middleware specification can be used to connect all the components without requiring implementation issues to be taken into account. Further, because it is a standard architecture for building robotic systems, individual components developed in different institutes can easily be connected.
Provide the required functions in a reconfigurable manner: We implement various functions from audio signal processing to dialog management in a uniform and reconfigurable manner. The developer can develop the entire system at a comparatively less development cost. In addition, the system can easily be adapted to different environments for realizing accurate recognition.
Figure 11 illustrates the overall architecture of the components provided in OpenHRI. The software covers all the functions for the development of the communication system and also incorporates an interface for establishing connections with other components that can provide multi-modal information.
The component architecture of our software is based on RT-Middleware. RT-Middleware is a middleware architecture for robotic applications that has been standardized by the OMG.
In RT-Middleware, each function of the robot is implemented as a "node." An application system can be developed by selecting the required components and connecting them to each other (the connections are called "links"). Figure 12 shows the "RT-SystemEditor" development tool to edit the links between the components.
In the specification of RT-Middleware, a "data port" is defined as a connection point of a link that realizes the transmission of a data stream. A "service port" is defined as an entry point of

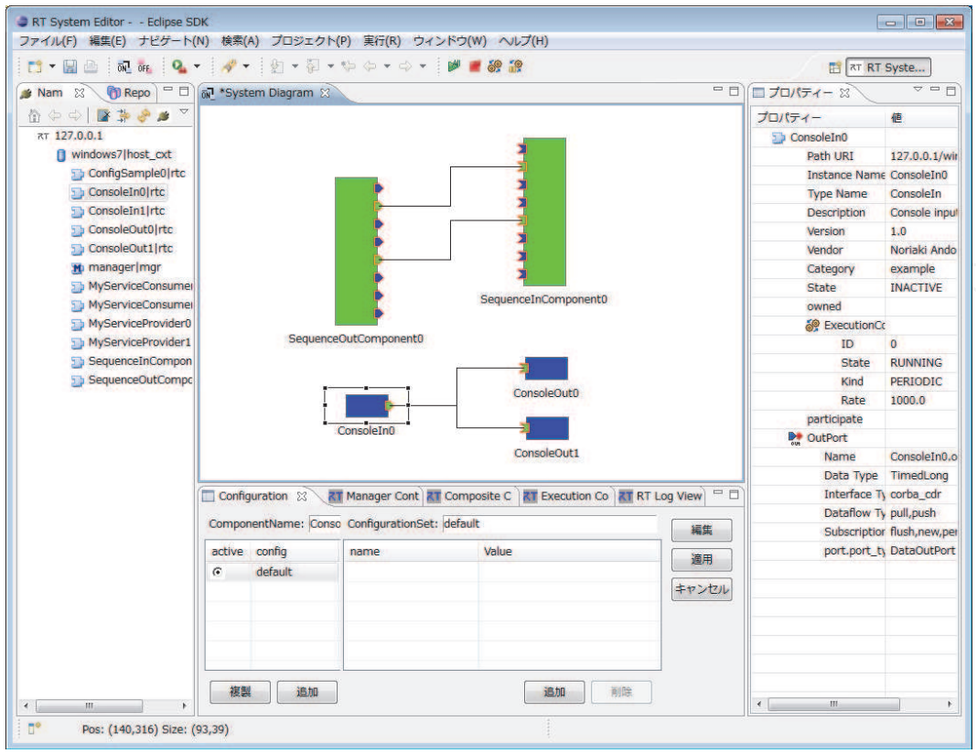


Fig. 12. Screenshot of RT-SystemEditor.

each function call for a service function. "Configuration parameters" are defined to configure each component.

OpenRTM-aist is an implementation of the RT-Component specification that supports C++, JAVA, and Python languages. It runs on various platforms such as Windows, Linux, Mac OS, and FreeBSD.

3.1.1 Audio input/output components

Audio input components accept the audio information from the sound device as input, convert it to a OpenRTM-aist data stream, and pass on the output to other linked components. These components also accept audio data streams from other components as input and pass on the output to the sound device.

We use portaudio, a cross platform audio input/output library, to implement both the audio input and output components. The components support both Windows and Linux platforms from monoral input to multichannel inputs.

3.1.2 Audio filter components

Audio filter components contain input and output ports.

The "sample rate conversion component" converts the sample rate of the audio stream by using an up/down sampling algorithm. The "echo cancel component" has two input streams; it subtracts the input of stream 1 from that of stream 2 by finding a maximum correlation. The "emphasis component" applies a signal processing algorithm to enhance or de-enhance the magnitude of the specified frequency in the data stream.

3.1.3 Voice recognition and synthesis components

The voice recognition component is based on Julius (Kawahara et al, 2000) in combination with English and Japanese acoustic models. Our component is designed to possess the following features: (a) the ability to read grammar format in W3C-SRGS XML form and (b) the ability to output the recognized result as an extensible XML stream.
Figure 13 shows an example of voice recognition grammar specified using W3C-SRGS format. Voice recognition grammar can be visualized by combination of "srgstojulius" and "juliustographviz" tool.
The voice synthesis component is based on Festival for English and Open_JTalk for Japanese. The component accepts plain text as input and provides a data stream in the form of a synthesized voice as output.

3.2 Incremental script development

Commands given by the human to the robot are diverse. The following are the factors that cause this diversity.
The nature of language: Human language is ambiguous, and different expressions can be used to give instructions that carry the same meaning.
Tasks: Robots working in a life environment have to accept a variety of tasks. In order to cope with this, it is necessary for them to understand a variety of commands.
Ability of the robot itself: The diversity is also caused by the ability of the robot itself. A command from a human becomes effective due to the functions of the robot. For example, humans do not say "walk N steps" to a robot on wheels.
The language comprehension system of the robot must be able to deal these diversities.
In the script-based development approach, diversity has been dealt with by stacking a newly developed script onto the existing scripts. By accumulating a number of scripts, the developer can accumulate the number of commands that the system can dealt with.
Incremental development of the state-transition model has previously been conducted using the "extension-by-connection" method (described in the next section). In this section, we propose an "extension-by-unification" method that can cope with the diversities mentioned above (described in Section 3.2.4).

3.2.1 Formalization of state-transition model

A state-transition model is a modeling method in which the input and output of the system assume the following form:

$$A := < I, S, O, \gamma, \lambda, s_0 > \tag{1}$$

where I represents the input alphabet, O represents the output alphabet, S represents the internal states, γ represents the state transition function, λ represents the output function and s_0 is the initial state.
The state transition function γ is defined in association with the state to the input.

$$\gamma : S \times I \rightarrow S \tag{2}$$

The output function λ is defined in association with the state to the input.

$$\lambda : S \times I \rightarrow O \tag{3}$$

```
<?xml version="1.0" encoding="UTF-8" ?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
         xml:lang="en"
         version="1.0" mode="voice" root="main">
  <rule id="main">
    <one-of>
      <item><ruleref uri="#greet" /></item>
      <item><ruleref uri="#command" /></item>
    </one-of>
  </rule>
  <rule id="greet">
    <one-of>
      <item>hello</item>
      <item>bye</item>
    </one-of>
  </rule>
  <rule id="command">
    <one-of>
      <item>pick</item>
      <item>give me</item>
    </one-of>
    <one-of>
      <item>apple</item>
      <item>cake</item>
      <item>remote</item>
    </one-of>
  </rule>
</grammar>
```

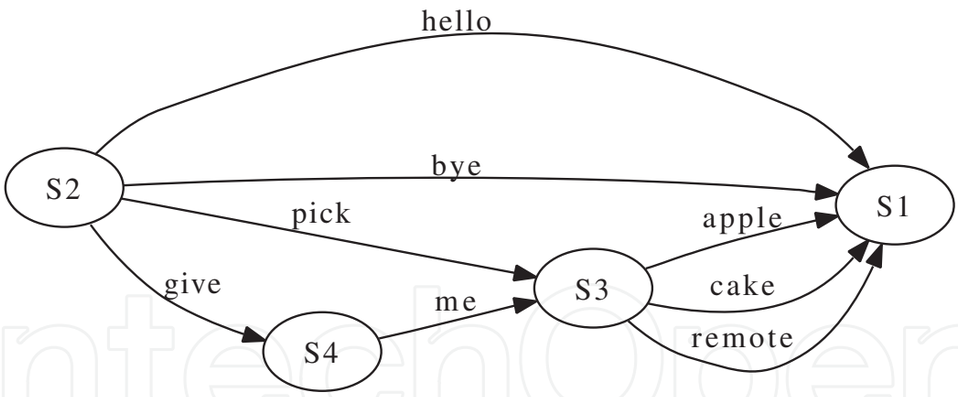


Fig. 13. Example of the voice recognition grammar and its visualization. The grammar is in W3C-SRGS form. “one-of” indicates the grammar matches either one of the child items. “ruleref” indicates reference to “rule” identified by the “id”.

When the system is in state s_t and get input alphabet i_t, state transition to s_{t+1} will occour as follows:

$$s_{t+1} = \gamma_{s_t, i_t} \tag{4}$$

At the same time, we get output alphabet o_t as follows:

$$o_{t+1} = \lambda_{s_t, i_t} \tag{5}$$

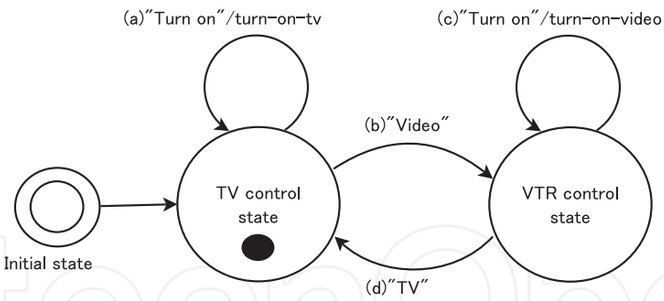


Fig. 14. Example of state-transition model.

Even the input to the system is same, the output of the sytem may be different, because the internal state s_t will be updated each time the system gets the input.
We have explained the stat-transiton model in an equation form, however, the state-transition model can be also presented in a 2-dimensional diagram called "state-transition diagram." In the diagram, each state is represented by a circle, and the transition between states is represented by arrows. In this chapter, we annotate the transition conditions and the associative actions by including text over each arrow. We use a black circle (called a "token") to represent the current state.
For example, Figure 14 represents a conversation modeled by the state-transition model.
The model presented in Figure 14, the initial state of the system is in "TV control" state. When the model gets the instruction "Turn on" as an input, it will output the command "turn-on-TV" and state transition "(a)" will occur. Then the token turns back to the same "TV control" state. When the model gets the instruction "Video" as an input, state transition "(b)" will occur and the token will move to "VTR control" state. This time when the instruction "Turn on" is given, state transition "(c)" occur and output the command "turn-on-video". In this way, we can model the context by defining an appropriate state and state transitions between the states.
Above example is expressed as follows in the equation form:

$$A := < I, S, O, \gamma, \lambda, s_0 > \tag{6}$$

$$I = (\text{"Turn on"}, \text{"Turn off"}, \text{"TV"}, \text{"Video"}) \tag{7}$$

$$S = (\text{"tv-control"}, \text{"vtr-control"}) \tag{8}$$

$$O = (\text{"turn-on-tv"}, \text{"turn-on-video"}, \text{"turn-off-tv"}, \text{"turn-off-video"}) \tag{9}$$

$$\gamma = \begin{pmatrix} s_0 & s_0 & s_0 & s_1 \\ s_1 & s_1 & s_0 & s_1 \end{pmatrix} \tag{10}$$

$$\lambda = \begin{pmatrix} o_0 & o_2 & none & none \\ o_1 & o_3 & none & none \end{pmatrix} \tag{11}$$

As we have seen here, the expression in equation form has an advantage in formalization, while the expression in diagram form has an advantage in quick understanding. In later

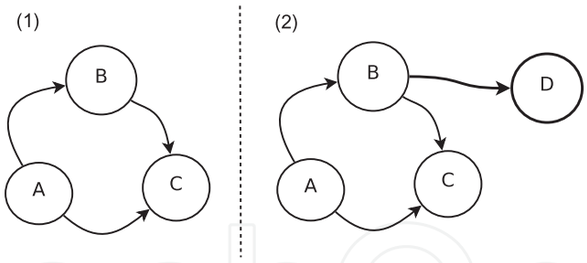


Fig. 15. Extension of a state machine using the extension-by-connection model.

discussion, we will use both the equation and the diagram forms to explain the concept quickly and formally.
State-transition model is very simple but very powerful modeling method and has been applied to very wide applications. Because the structure of state-transition model is very simple, it is frequently misunderstood that the state-transition model can only model simple behavior. However, it can model diverse behavior by applying some extensions (e.g. Huang et al (2000), Denecke (2000)).

3.2.2 Extension-by-connection method

The simplest way to extend state-transition model is as follows.

1. Add a new state to the existing state-transition model.
2. Add a new transition from the existing state to the new state.

This process is illustrated in Figure 15.
Here, we formulate the above process. Let the existing state-transition model be A and the accumulated state-transition model be A'.
As explained in Section 3.2.1, the existing state-transition model A can be represented by following form.

$$A :=< I, S, O, \gamma, \lambda, s_0 > \tag{12}$$

Here, S is the set of state $s \in S$. The transition function γ can be defined in any form. In this chapter, we use the matrix of $S \times I$, in which the transition from state s_t to state s_{t+1} can occur if $\gamma_{S_t,i} = s_{t+1}$.
Similarly, we define the accumulated state-transition model A' as follows:

$$A' :=< I, S', O, \gamma', \lambda', s'_0 > \tag{13}$$

Then, the new state ΔS can be calculated as follows:

$$S' = S \cup \Delta S \tag{14}$$

Here, $S' \cap \Delta S = \emptyset$.
The new state transition $\Delta\gamma$ can be calculated as follows:

$$\gamma'_{s_t,i} = \gamma_{s_t,i} \qquad (s_t \in S, i \in I) \tag{15}$$

$$\gamma'_{s'_t,i} = \Delta\gamma_{s'_t,i} \qquad (s'_t \in S', i \in I) \tag{16}$$

$$\lambda'_{s_t,i} = \lambda_{s_t,i} \qquad (s_t \in S, i \in I) \tag{17}$$

$$\lambda'_{s'_t,i} = \Delta\lambda_{s'_t,i} \qquad (s'_t \in S', i \in I) \tag{18}$$

The transition function of the accumulated part $\Delta\gamma$ needs to be defined based on the transition from the existing state S . Therefore, $\Delta\gamma$ will be a matrix of $S' \times I$. Note that the new state ΔS can be expressed only by the newly defined part, but the transition of the accumulated part $\Delta\gamma$ includes both old state S and new state ΔS in its definition.
The state-transition model is easy to understand in drawing a state-transition diagram. Extension-by-connection can also be carried out very easily by editing this diagram. There are several GUI that can add state-transition rules through the operation of mouse clicks.

3.2.3 Problems with the extension-by-connection method

Extension-by-connection is a useful method, but it has the following problems.
As we can see in Equation 14 and Equation 16, the definition of $\Delta\gamma'$ requires both S and ΔS. This causes problems in the function development of robots. For example, let us consider the following scenario:

1. Robot "A" has function A, and we have already developed a state-transition model $A^A$to realize the function.
2. For the robot "A" to accumulate function C, we have extended the state-transition model to A^{AC}.
3. We have developed another robot, "B," which has function B. And we want to add function C to this robot.

Here, the state-transition model for function C is already developed for robot A. We want to reuse the model for robot B. Here, we discuss whether such a diversion would be possible.
First, the state S^{AC} is easily separable from state S^A and state S^C:

$$S^C = S^{AC} - S^A \tag{19}$$

However, the definition of state-transition function γ^{AC} is as follows:

$$\gamma^{AC}_{s^A_t,i} = \gamma^A_{ij} \qquad (s^A_t \in S^A, i \in I) \tag{20}$$

$$\gamma^{AC}_{s^{AC}_t,i} = \gamma^C_{ij} \qquad (s^{AC}_t \in S^{AC}, i \in I) \tag{21}$$

γ^C contains state S^A in its definition.
Because states S^A and S^B are defined for different types of robots, A and B are not equal. In addition, because the transition for the function C is defined dependently on state S^A, we cannot replace variables like $S^{AC} = S^{BC}$, which means that we cannot use γ^C to extend the state-transition model A^B. The state transition of function C developed for robot A cannot be diverted for the extension of robot B.
Ideally, once a feature is developed, it would be possible to share with other robots that need the same feature. In order to achieve this, we introduce the extension-by-unification method.

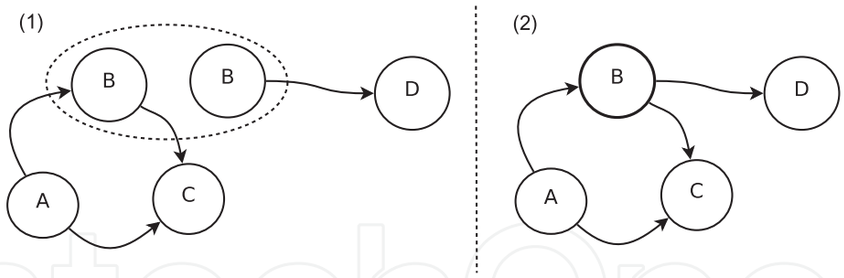


Fig. 16. Extension of the state-transition model using the extension-by-unification method.

3.2.4 Extension-by-unification method

In the extension-by-unification method, we extend the state-transition model by the following procedure:

1. Develop a state-transition model to realize a new function.
2. Unify a state with the same ID between the existing and the new state-transition models.

This process is illustrated in Figure 16.
Here, we formulate the above process.
The existing state-transition model A can be represented by state S, state transition γ, and initial state s_0:

$$A :=< I, S, O, \gamma, \lambda, s_0 > \tag{22}$$

Similarly, the new state-transition model A' is represented as follows:

$$A' :=< I, S', O, \gamma', \lambda', s'_0 > \tag{23}$$

We accumulate the state-transition model A'' by unifying A and A'. First, we calculate state as follows:

$$S'' = S \cup S' \tag{24}$$

Here, $S \cap S' \neq \emptyset$.
Next, the transition between the state S'' is calculated as follows:

$$\gamma''_{s_t,i} = \gamma_{s_t,i} \qquad (s_t \in S, i \in I) \tag{25}$$
$$\gamma''_{s'_t,i} = \gamma'_{s'_t,i} \qquad (s'_t \in S', i \in I) \tag{26}$$

$$\lambda''_{s_t,i} = \lambda_{s_t,i} \qquad (s_t \in S, i \in I) \tag{27}$$
$$\lambda''_{s'_t,i} = \lambda'_{s'_t,i} \qquad (s'_t \in S', i \in I) \tag{28}$$

By defining initial state s''_0 to be $s''_0 = s_0$, the extended state-transition model A'' will be as follows:

$$A'' =< I, S'', O, \gamma'', \lambda'', s''_0 > \tag{29}$$

As visible in Equation 26, the transition function γ' is an $S' \times S'$ matrix that only includes state S' in its definition. The extension-by-unification method does not require the definition of the original state in the accumulated part of the state-transition model.
As noted in Section 3.2.3, in the conventional extension-by-connection method, the definition of the accumulated part of the state-transition model depends on information on the existing state. It is limited in terms of reusing scripts for this reason. The proposed extension-by-unification method does not have this problem. Using this method, we can significantly increase the reusability of the state-transition model.

3.2.5 Visualization of unifiable states

By using the above algorithms, the possibility of unification between scripts can be identified as "Unifiable," "Unifiable (Occurrence of isolated state)," or "Conflict". Similarly, scripts can be classified as "Executable" or "Unexecutable." By comparing a script and an adaptor definition for the existing scripts, we can obtain a list of scripts annotated with 6 (3×2) classes. Our script management server displays the above list at the bottom of each wiki page. By displaying the list, the developer can easily find a script that can be included in his/her current application.
Figure 17 shows overview of the editing system and Figure 18 shows example of using the web based interface.

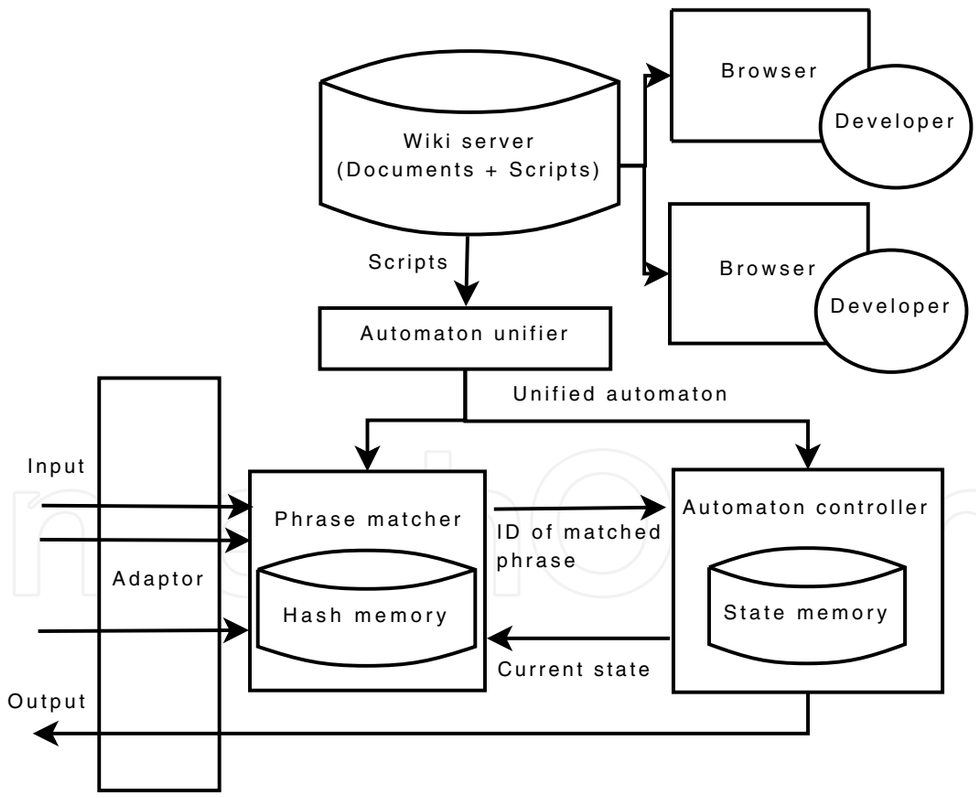
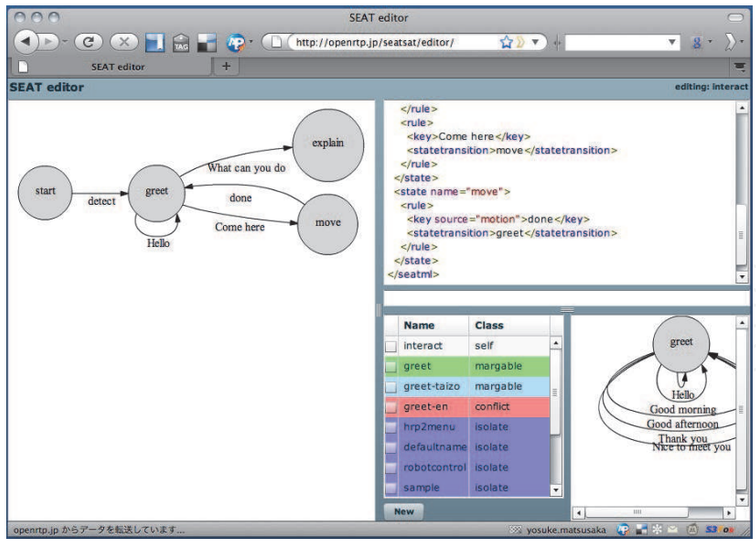
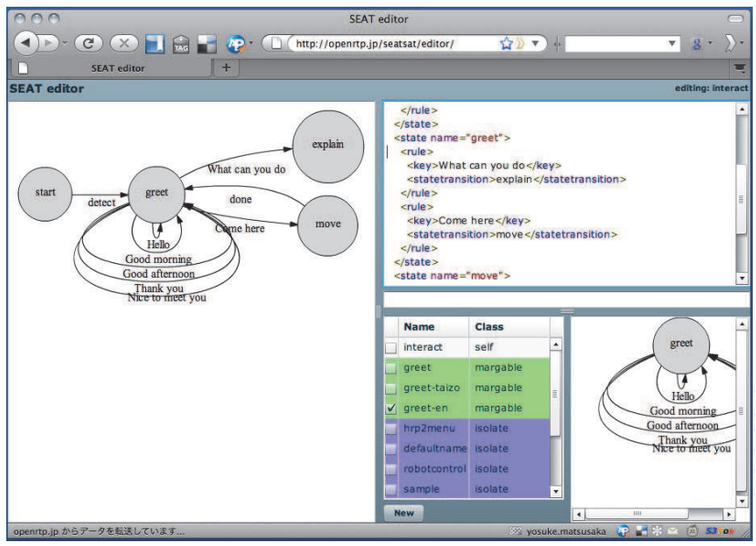


Fig. 17. Overview of the editing system.



a) Overview of the development interface. Visualization of the state-transition model (left), XML based editing panel (right top), real-time annotation of existing scripts (right bottom). Editing task script. When the developer types the keyword “Hello”, the existing script from the script database is annotated as “conflict” and suggest to reuse. At this step, the system only accepts 3 (“Hello”, “What can you do?”, “Come here”) phrases.



b) When the developer check the “greet” script, which already contains several vocabulary for greeting, it is unified to the task script. As a result, the developer only had to increment the application specific vocabulary to realize the whole script with many vocabularies. At this step, the system accepts 7 (“Hello”, “Good morning”, “Good afternoon”, “Thank you”, “Nice to meet you”, “What can you do?”, “Come here”) phrases.

Fig. 18. Example of using the web based interface.

4. Summary

In this chapter, we have introduced our health exercise robot TAIZO and the development background. TAIZO has the function to accept commands by voice and keypad, and demonstrates by using its body in collaboration with human demonstrator. Through the experiment, we have measured not only the error rate of both voice and key inputs, but also, the demonstrative effect of using each method. Through real demonstrations, we have confirmed that the robot is effective for giving the incentive to engage in health exercises.
We have also discussed about some practical problems to make difficult the development of communication function for humanoid robot. By introducing the component based architecture and extension-by-unification method to develop the dialog script, scripts created in the past can easily be reused in the new application. In the conventional extension-by-connection method, the developer had to develop each function in turn, because it did not support the "merging" of scripts that had been developed simultaneously.
We believe, in future, how advanced the computer graphics are, humanoid robots still keep its strongness to affect the human to move. We hope such powerful medium would be used more in the feature, by developing practical techniques as described in this chapter.

5. References

Y.Matsusaka, H.Fujii, I.Hara: Health Exercise Demonstration Robot TAIZO and Effects of Using Voice Command in Robot-Human Collaborative Demonstration, Proc. IEEE/RSJ International International Symposium on Robot and Human Interactive Communication, [in print] (2009)

Y. Moon. (2001). Sony AIBO: The World's First Entertainment Robot," in The Harvard Case Collection, Harvard Business School Publishing

K. Wada, T. Shibata, T. Saito and K. Tanie. (2002). Robot Assisted Activity for Elderly People and Nurses at a Day Service Center," in Proceedings of the IEEE International Conference on Robotics and Automation, pp.1416-1421

Y. Matsusaka, "History and Current Researches on Building Human Interface for Humanoid Robots," in Modeling Communication with Robots and Virtual Humans, Lecture Notes in Computer Science, Springer, 2008, pp.109–124.

K. Ganeshan, "Introducing True 3-D Intelligent Ed-Media: Robotic Dance Teachers," in Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006, pp.143–150, 2006.

Y. Katagiri, T. Takahashi, Y. Takeuchi, "Social Persuasion in Human-Agent Interaction,", in Proceedings of Second IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2001, pp.64–69, 2001.

K. Hayashi, T. Kanda, T. Miyashita, H. Ishiguro and N. Hagita, "ROBOT MANZAI -Robot Conversation as A Passive-Social Medium-," International Journal of Humanoid Robotics, 5(1), pp.67-86, 2008.

H. Hirukawa, F. Kanehiro, S. Kajita, "OpenHRP: Open Architecture Humanoid Robotics Platform," in Tracts in Advanced Robotics, Springer, pp.99–112, 2003.

SEAT,SAT: AIST-OpenRTP Project, http://openrtp.jp/seatsat

N.Iwahashi: Language Acquisition Through a Human-Robot Interface, In Proc. ICSLP-2000, vol.3, pp. 442–447 (2000)

D.Roy: Grounded Spoken Language Acquisition: Experiments in Word Learning, IEEE Transactions on Multimedia. vol.5, no.2, pp. 197–209 (2003)

A.Symeonidisa, I.Athanasiadisb and P.Mitkasa: A Retraining Methodology for Enhancing Agent Intelligence, Knowledge-Based Systems, vol.20, issue.4, pp. 388–396 (2007)

T.Winograd: Understanding Natural Language, Academic Press (1972)

R.Brooks: Intelligence Without Representation, Artificial Intelligence, vol.47, pp. 139–159 (1991)

L.Kaelbling: A Situated-Automata Approach to the Design of Embedded Agents, ACM SIGART Bulletin, vol.2 , issue.4, pp. 85–88 (1991)

B.Yartsev, G.Korneev, A.Shalyto, V.Kotov: Automata-Based Programming of the Reactive Multi-Agent Control Systems, Proc. IEEE International Conference on Integration of Knowledge Intensive Multiagent Systems, pp. 449–453 (2005)

T.Kanda, H.Ishiguro, T.Ono, M.Imai, R.Nakatsu: Development and Evaluation of an Interactive Humanoid Robot Robovie, Proc. IEEE International Conference on Robotics and Automation, pp. 1848–1855 (2002)

T.Kanda, M.Shiomi, Z.Miyashita, H.Ishiguro, and N.Hagita: An Affective Guide Robot in a Shopping Mall", Proc. ACM/IEEE International Conference on Human-Robot Interaction, pp. 173–180 (2009)

F.Huang, J.Yang, A.Waibel: Dialogue Management for Multimodal User Registration, In Proc. Int'l Conf. on Spoken Language Processing, Vol.3, pp. 37–40 (2000)

M.Denecke: Informational Characterization of Dialogue States, In Proc. Int'l Conf. on Spoken Language Processing, Vol.2, pp. 114–117 (2000)

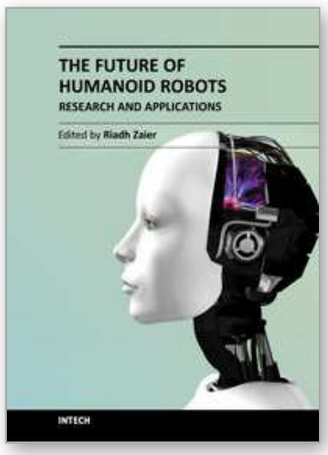
Voice Extensible Markup Language (VoiceXML) Version 2.0: http://www.w3.org/TR/voicexml20/

N.Ando, T.Suehiro, K.Kitagaki, T.Kotoku, W.Yoon: RT-Middleware: Distributed Component Middleware for RT (Robot Technology), Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3555–3560 (2005)

T. Kawahara, A. Lee, T. Kobayashi, K. Takeda, N. Minematsu, S. Sagayama, K. Itou, A. Ito, M. Yamamoto, A. Yamada, T. Utsuro and K. Shikano: Free Software Toolkit for Japanese Large Vocabulary Continuous Speech Recognition, In Proc. Int'l Conf. on Spoken Language Processing, Vol. 4, pp. 476–479, (2000)

I.Hara, F.Asano, H.Asoh, J.Ogata, N.Ichimura, Y.Kawai, F.Kanehiro, H.Hirukawa, K.Yamamoto: Robust Speech Interface Based on Audio and Video Information Fusion for Humanoid HRP-2, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2404–2410 (2004)

F.Asano, K.Yamamoto, I.Hara, J.Ogata, T.Yoshimura, Y.Motomura, N.Ichimura, H.Asoh: Detection and Separation of Speech Event Using Audio and Video Information Fusion and Its Application to Robust Speech Interface, EURASIP Journal on Applied Signal Processing, vol.2004, issue.11, pp. 1727-1738 (2004)



The Future of Humanoid Robots - Research and Applications

Edited by Dr. Riadh Zaier

ISBN 978-953-307-951-6
Hard cover, 300 pages
Publisher InTech
Published online 20, January, 2012
Published in print edition January, 2012

This book provides state of the art scientific and engineering research findings and developments in the field of humanoid robotics and its applications. It is expected that humanoids will change the way we interact with machines, and will have the ability to blend perfectly into an environment already designed for humans. The book contains chapters that aim to discover the future abilities of humanoid robots by presenting a variety of integrated research in various scientific and engineering fields, such as locomotion, perception, adaptive behavior, human-robot interaction, neuroscience and machine learning. The book is designed to be accessible and practical, with an emphasis on useful information to those working in the fields of robotics, cognitive science, artificial intelligence, computational methods and other fields of science directly or indirectly related to the development and usage of future humanoid robots. The editor of the book has extensive R&D experience, patents, and publications in the area of humanoid robotics, and his experience is reflected in editing the content of the book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen