Autonomous Behavior Control Architecture of Entertainment Humanoid Robot SDR-4X

Conference Paper · December 2003

DOI: 10.1109/IROS.2003.1250752 · Source: IEEE Xplore

Masahiro Fujita
Sony Corporation

Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Autonomous Behavior Control Architecture of Entertainment Humanoid Robot SDR-4X

Masahiro Fujita, Yoshihiro Kuroki, Tatsuzo Ishida, and Toshi T. Doi
Intelligent Dynamics Laboratory
Sony Corporation
Email: mfujita@pdp.crl.sony.co.jp

Abstract

In this paper we describe the autonomous behavior control architecture of SDR-4X, which serves to integrate multi-modal recognition and motion control technologies. We overview the entire software architecture of SDR-4X, which is composed of perception, short and long term memory, behavior control, and motion control parts. Regarding autonomous behavior control, we further focus on issues such as spontaneous behavior generation using a homeostasis regulation mechanism, and a behavior control/selection mechanism with tree-structured situated behavior modules. In the autonomous behavior control architecture, we achieve three basic requirements, which are the concurrent evaluation of the situation of each behavior module, concurrent execution of multiple behavior modules, and preemption (behavior interruption/resume capability). Using the autonomous behavior control architecture described, we demonstrate that SDR-4X can spontaneously and passively interact with a human.

I. INTRODUCTION

We have proposed Robot Entertainment as a new application of autonomous robots [1]. The aim of this proposition is to establish a new industry with autonomous robots and artificial intelligence. We previously reported the development of a quadruped robot, AIBO, for pet-type applications [2], and a biped robot SDR-3X as a motion entertainment prototype with dancing performance [3][4]. We further developed a biped robot SDR-4X as a prototype partner robot[5], which emphasizes human interaction capability more than SDR-3X.

One of the significant technical achievements of SDR-4X is its integration of many advanced technical features, such as real-time motion control, face recognition, speech recognition, and so on. A well-designed architecture is a key to integrating these technologies. Moreover, in general, the architecture is very important to develop many applications for motion and communication entertainment. In our case, a main goal of SDR-4X is to build a partner robot, which has to behave spontaneously as well as to listen to spoken human voice commands. Another possible application of SDR-4X is as a “remote control robot”, which is controlled by a joystick of a PC. From an architectural point of view, the architecture for a partner robot is more general than that required for a “remote control robot”.

In this paper, we describe our autonomous behavior control architecture, by which behaviors are properly selected depending on the situation of the external world’s conditions and simulated internal instincts. In the following section, we describe an overview of SDR-4X. We then describe the logical architectural structure of the entire software architecture of SDR-4X, followed by a description of various functional requirements for the autonomous behavior control architecture. Then we present some preliminary experimental results of our implementation.

II. OVERVIEW OF SDR-4X

Fig. 1 shows the basic configuration of SDR-4X, with a 58 cm height and 6.5 kg weight, and having 28 DOF in its major joints. Each leg has 6 DOF, the trunk has 2 DOF, each arm has 5 DOF, and the neck has 4 DOF. In addition, five independent driving fingers are also attached to each hand. Fig. 2 shows the basic specifications of SDR-4X.

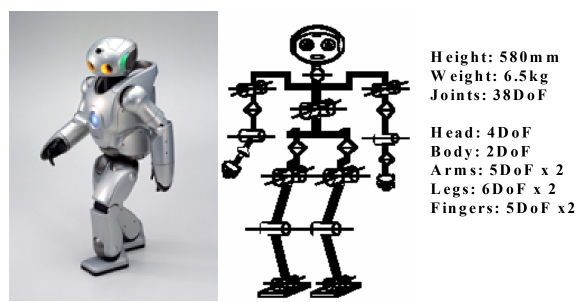

Fig. 1 : Basic Configuration of SDR-4X

CPU	64 bit RISC x 3
Memory	64 MB DRAM x3
OS & Architecture	Aperios & OPEN-R
Program Media	Memory Stick
Extention Bus	PC Card Slot
Image Input	110,000pixels color CCD x 2
Sound Input/Output	7 microphones
Walking Speed	6m/min, max (unleveled surface)
	20m/min. max (flat surface)
Ability of Adaptive Walki	approx. 10mm uneven surface
(on non-slip condition)	Approx. 10deg. tilt surface)
Weight (including battery	Approximately 6.5kg
Dimension	Approx. 680mm

Fig. 2: Basic Specification of SDR-4X

Note that we increased the number of CPUs from two to three in this new model, to increase computational resources for speech recognition and synthesis. It is then possible to implement a standalone SDR-4X with Large Vocabulary Continuous Speech Recognition (LVCSR). Increasing the CPU system is easily achieved by the OPEN-

R multi-system core architecture drawn in Fig. 3. We call the system core as aSDR, iSDR, and mSDR: the aSDR is used for speech recognition and speech synthesis; the iSDR is used for vision, Short-Term-Memory (STM), Long-Term-Memory (LTM), Internal State Module (ISM), and Behavior Control Module (BCM); and the mSDR is used for real-time motion control.

For the head component, there are two CCD camera for the stereo-vision system, and 7 microphones for the sound processing system. We use Field-Programmable-Gate-Array (FPGA) for stereo-vision computation and DSP for sound signal processing including sound localization. Fig. 4 shows a picture of the head of SDR-4X.

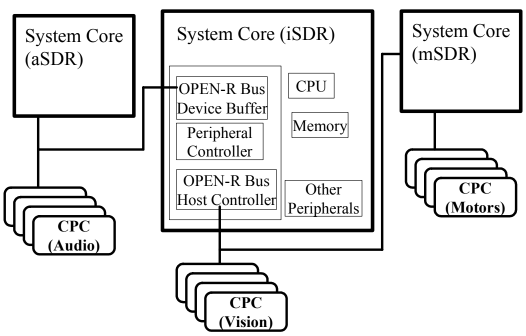

Fig. 3: Multi-System Core Architecture by OPEN-R

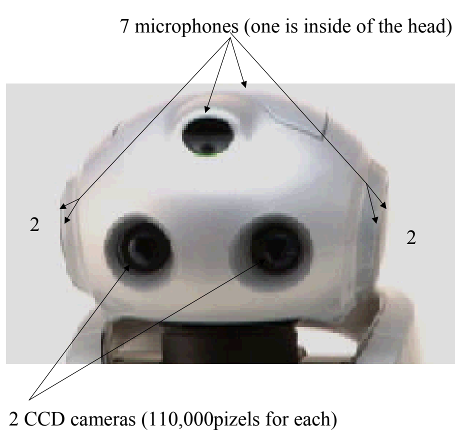

Fig. 4: The head of SDR-4X with a stereo vision system and an audio signal processing with multiple microphones.

III. Logical Architecture for Autonomous Behavior Control

A. Requirements

In order to develop a robot platform for a partner robot, we have to emphasize the dialogue capability when compared with SDR-3X or AIBO's earlier architectures [2]. This goal requires us to improve our software architecture not only to simply add a dialogue module, but also to provide other essential functional modules such as a "memory" capability. We consider that the following functions are necessary for our software architecture to develop a partner robot.

1) Spontaneous behavior

We prefer a robot that behaves spontaneously. Namely, we would like to build a robot that can spontaneously talk to a user, take a rest, and search for something, e.t.c... In addition, if a user would like to talk with the robot, basically the robot should respond to the user. However, it could also be acceptable that the robot sometimes refuses to respond to the human's request. How often the robot refuses or accepts a command can be considered as part of the personality of a robot. In this paper, we will not discuss personality but rather the underlying architecture for how to build such behavioral tendencies.

2) Reflexive behavior and deliberate behavior

It is very important for a life-like robot to respond very quickly to threat stimuli. In addition, it is also very important for such a robot to track an object such as a moving face of a talking user for good interaction. On the other hand, a partner robot with some level of intelligence must behave using deliberate planning such as path planning. How to integrate reflexive and deliberate behavior is a key issue for our autonomous behavior control architecture.

3) Spatial Perception with memory

Since the vision sensor has a limited view angle, it is impossible to observe simultaneously the multiple objects whose locations are distributed wider than the view angle. Then, the robot need the capability of memorizing the objects' spatial locations, so that the robot can behave intelligently with objects naturally distributed in a real-world setting.

4) Dialogue with memory

It is still difficult for a robot to understand the meaning of a human's spoken language. However, there are some methods for a robot to vocally respond to human's utterances. One such method is to utilize some words that a human previously said. If the robot uses these memorized words after a while in the proper situation, the human feels there is some intelligence in the robot.

B. Approach

To achieve the requirements described above, we take the following approach:

1. *Homeostasis Regulation for spontaneous behavior*
2. *Layered Architecture for integration of reflexive and deliberative behaviors*
3. *Short Term Memory (STM) for spatial perception.*
4. *Long Term Memory (LTM) for dialogue with memory*

In the next subsection, we describe the entire software architecture including an explanation of each of these areas.

C. Implementation

Fig. 5 shows the logical architecture for the autonomous behavior control architecture of SDR-4X. Roughly speaking, it is divided into 5 parts:

1. Perception
2. Memory
3. Internal state model
4. Behavior control
5. Motion control

In this paper, we mainly describe the Memory, Internal state model, and Behavior control parts. For the others, please refer to [5]. Furthermore, regarding autonomous behavior control, the Internal state model and the Behavior control part are the key to implementing the architecture. In this section, we describe the functional role of each part, and in the following sections we explain these parts from the autonomous control architectural point of view.

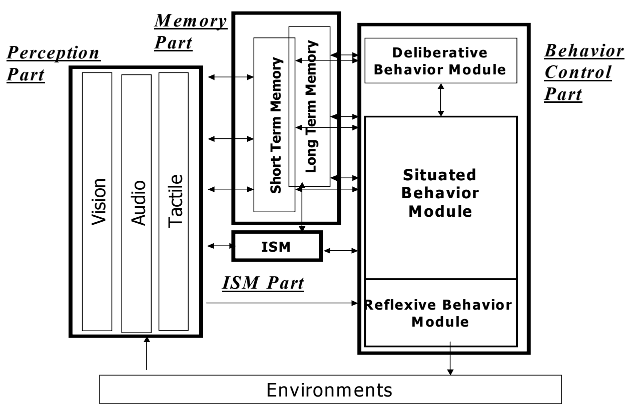

Fig. 5: Logical architecture for autonomous behavior

1) Perception Part

In the perception part, there are three sensor channels, which are vision, audio, and tactile sensors. In visual perception, there are two major functions, which are to detect a floor to walk while avoiding obstacles, and to detect and identify human faces. In the audio channel, speech processing is the main function, which includes sound direction estimation, sound event identification, and Large Vocabulary Continuous Speech Recognition (LVCSR). In the tactile sensor channel, the signals from touch sensors are processed and identified as several types of touching such as “hit” and “pat”.

2) Memory part

In the memory component, there are two sub-modules, which are Short Term Memory (STM) and Long Term Memory (LTM). For the requirement of spatial perception with memory, we implement STM. From sound information, the robot can know the source direction, and from the results of face detection, the robot can know the face’s direction and approximate distance (assuming that the face size is roughly known). Using this detected information and determining its position from the robot’s kinematics, SDR-4X can compute its position relative to the detected humans and objects. This position is then memorized in STM. Using the information of STM, the robot can handle tasks involving several humans and objects, which are located outside of the limited view range.

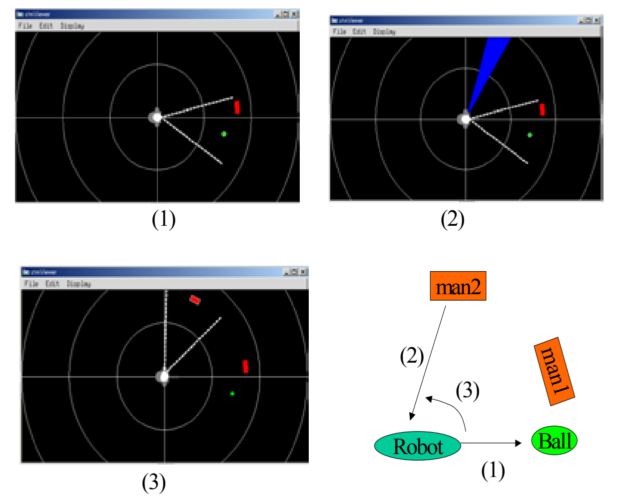

Fig. 6: Short Term Memory (STM) Viewer:

Fig. 6 explains the function of STM. The top left figure shows the case when a robot looks at a ball (green circle). We use information regarding the size of the ball as well as an average of humans’ face size, therefore the distance can be determined. The dot-lines show the camera’s view angle. Inside the view range, there is a person (man1) whose face is detected. The top right figure shows the situation where a voice signal now comes from the left-hand-side of the robot, which is out of the camera’s view range, and at an unknown distance (shown by the colored fan-shape). The bottom left figure shows that the robot turns to the voice signal direction and finds man2, whose face is detected, and then his distance is determined. The robot still remembers the position of the ball and the man1.

Regarding LTM, there are two types of LTM used. One is an associative memory and the other is frame-type (symbol) memory. The associative memory is mainly used to remember an identified face, an identified voice, and an acquired name. Since the face learning, speaker learning, and unknown word acquisition are carried out statistically, we can use neural networks to implement the associative memory. Using the associative LTM, SDR-4X can identify the user either from its face or voice. Then, SDR-4X can recall the experiential memory with associated with the particular user. Using this experiential memory, SDR-4X can behave differently towards each individual.

Regarding the experiential memory, we implemented a frame-type symbol memory, which is the second type of LTM used. For example, through conversations with a user, SDR-4X can build a frame-type memory for each user, which includes the user’s birthday, favorite items, and so on. These memories can be reused in the different situations.

Note that there is a fundamental problem in symbol processing called the Symbol Grounding Problem. The associative LTM can be considered to contain physically grounded symbols, which grounds it to the physical world via perceptual channels [6]. We further expand this idea to associate the physically grounded symbol with emotional information, and use the concept of an Emotionally Grounded Symbol, through which the robot can behave properly toward its acquired symbols[7][8]

3) Internal State Model part

In Internal State Model (ISM), various internal variables alter their values depending on the passage of time and incoming external stimuli. Basically, a behavior is selected in order to keep these internal variables within proper ranges. ISM is the core for spontaneous behavior generation and response to external stimuli, which is one of the basic requirements described in the previous section. We describe ISM in the following section in more detail.

4) Behavior Control Module part

There are three different behavior control modules (BCMs): the Reflexive Behavior Control Module, the Situated Behavior Control Module, and the Deliberative Behavior Control Module. The Reflexive Behavior Control Module depends on the configuration of SDR-4X, so that response to external stimuli can be carried out very quickly. Since it is easy to program rapid responses using mechanical configuration dependent equations, a portion of the reflexive behaviors are implemented in the motion control part described in the next subsection.

The Situated Behavior Control Module includes many situated behavior modules, each of which can be properly utilized for a particular situation based on external stimuli and ongoing internal drives. We developed this behavior module based on an ethological study[9]. The situated behavior module is relatively easy to develop, compared to behavior modules that can work in more general situation [10]. Therefore, it is easy for us to develop many situated behavior modules, so that the entire behavior module set covers many different situations.

Furthermore, we consider that dialogue can be performed by the situated behavior module architecture. This is implicitly suggested in the research program named the Talking Head Project[11], where many language games are implemented to acquire a language capability for synthetic agents including robots.

In our implementation, there are many dialogue-related behavior modules, each of which is activated when the corresponding situation is satisfied. We will describe the situated behavior module in more detail later.

In the Deliberative Behavior Control Module, computationally heavy tasks such as path planning are conducted. In the implementation, we use the same behavior modules for the deliberate behavior tasks. Then, complex tasks, such as the path planning process, are actually called from the situated behavior module, so that multiple tasks are processed in parallel in the Behavior Control Module.

5) Motion control part

We realized real-time integrated adaptive motion control and real-time gait pattern generation, enabling SDR-4X to walk on unbalanced terrain, adaptively move against an external force, fall down with shock absorbing motions, and recovery motions. There fast feedback is used in order to realize this real-time control, which can be considered similar to the reflexive behaviors previously described.

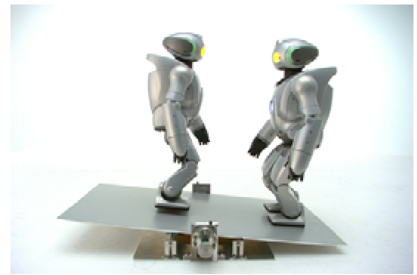

Fig. 7: Walking on unbalanced terrain.

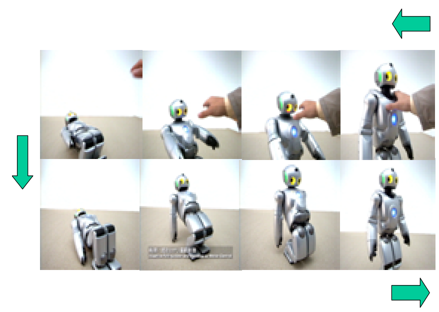

Fig. 8: Falling down and recovery motions

Autonomous behavior can be realized by integrating these parts. Among the components, ISM and BCM are key to realizing natural autonomous behavior generation. In the next section, we focus on the ISM and Behavior control parts.

IV. Regulating Internal State for Autonomous Behavior

In this section, we explain the role of ISM from the autonomous behavior control point of view. The basic mechanism of action selection for our behavior control model is to evaluate both external stimuli and ongoing internal drives. We employ the "homeostasis regulation rule" for action selection [9]. Namely, internal variables are specified that must be regulated and maintained within proper ranges. Behavioral actions and changes within the environment produce corresponding changes in these internal variables. The basic rule for action selection is to use the regulation of the internal variables as a motivational "drive" signal for the behavior modules. (Fig. 9)

For example, if one of the internal variables named "energy" is less than the regulation range, then some behavior modules that increase the "energy" receive a larger motivational drive signal, so that these modules are

activated for potential selection. If the module is selected, the resulting behavior produces an increase in "energy".
External stimuli also activate corresponding behaviors. For example, even if "energy" is within an acceptable range, if the robot observes a battery, then the "battery charging" behavior activation becomes high and active for potential selection. Thus, the homeostasis regulation mechanism naturally and properly selects behavior automatically.

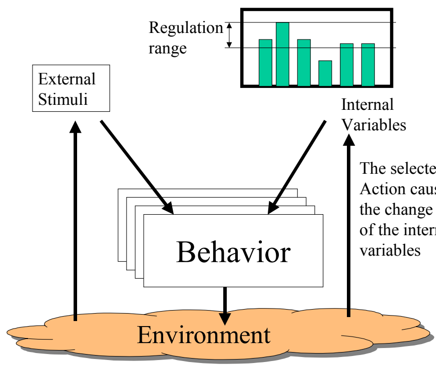

Fig. 9: Homeostasis Regulation for Action Selection

V. SITUATED BEHAVIOR MODULE CONTROL

In this section, we mainly explain the situated behavior control module from the overall Behavior Control Module discussed in the last section. Compared with the behavior-based architecture employed for systems such as AIBO, we improved the basic functions of the behavior module based on the following requirements.

A. Basic Requirements for Behavior Modules

There are three basic requirements for behavior control architecture.

1. *Concurrent Evaluation:* In order to behave properly in a real-world environment, we employ situated a behavior-based architecture[10][11], where a behavior module is designed for a particular situation. By building many such situated behaviors, we can achieve a robot that can behave in many different situations. It is then important for the behavior modules to monitor the situations concurrently, so that the proper behavior can be selected in a dynamically changing real-world environment. Thus, concurrent evaluation by many situated behavior modules is a basic requirement for our behavior control architecture.

2. *Concurrent Execution*: Assume that SDR-4X can read a newspaper with its right hand, and can drink a cup of coffee with its left hand, or in general, assume that a robot can execute a different task with each hand. Somehow the system needs to know what kind of mechanical resources are to be used to achieve the task objectives. It is necessary for our architecture to perform concurrent execution.

Preemption (Behavior Interrupt and Resume): Consider the situation in Fig. 6 (STM). Assume that first the robot talks first with man1, and then man2 calls from the left-hand-side direction. For this reason the robot has to turn towards man2 during the conversation with man1, and the behavior (conversation) should be stopped. In this situation, we need a control architecture that can stop a behavior and then resume it after finishing an emergency task. This is the requirement of behavior interrupt and resume.

B. Approach

Our approach to achieve the requirements are summarized as three items, (1) situated behavior module with situation evaluating and action execution functions, (2) resource management with a tree-structured architecture, and (3) state management of the situated behavior module.

1) Situated behavior module with evaluation function and action execution function.

First, we design two basic functions for the situated behavior module. One is the "monitor function" and another is an "action function". The monitor function is used for evaluating the situation and calculating an activation value, which indicates how much a given behavior is relevant in the current situation. The action function is used for executing the behavior. We employ a state machine for the action function. That is, depending on the state and input, the action function decides to output action commands.
The monitor function is periodically called by the behavior control system, so that the system always evaluates the situation for the situated behavior modules.

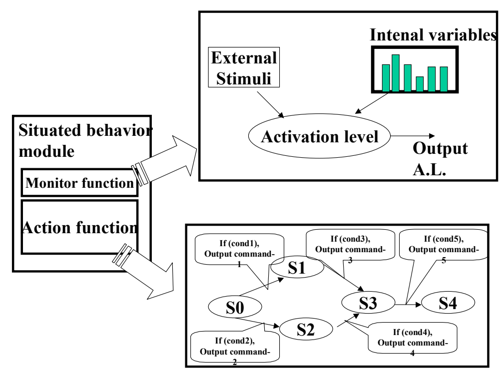

Fig. 10: Monitor and action functions in a situated behavior module

Note the monitor function evaluates both the external stimuli and internal variables. If we adjust some parameters of the monitor function, we can make several different personalities for SDR-4X. For example, if we put more weight on the external stimuli, then SDR-4X tends to select a behavior in response to the external stimuli but not to the internal drives. On the other hand, if we more weight on the internal drives, then SDR-4X tends to select a behavior depending more on SDR-4X's internal motivations, which looks like a "selfish" robot. In addition, if we control the parameters that assign the priority of the regulated internal variables, we can also design different personality types for SDR-4X. Thus, the architecture using the Homeostasis regulation mechanism with ISM can create many personalities for SDR-4X.

2) Resource Management with tree-structured architecture

In order to achieve the concurrent execution requirement, the behavior control system has to know what kind of resources are needed when the system selects for execution specific behavior modules. Then, when the system evaluates the behavior value, the behavior module also provides information regarding the required resources. Competition of activation values is done in the following order. First, the behavior with the highest activation value is selected and then if there still remain available resources, another competition is conducted between the behavior modules, which can work with the remaining resources. By continuing to select behavior modules until there remain no available resources, the behavior control system can execute multiple behavior modules with different resources in parallel.

The tree-structure is introduced to coordinate many behavior modules. Usually a behavior can be naturally divided into multiple sub-behaviors. For example, as shown in Fig. 11, a "dialogue behavior module" can be decomposed into "search the target (a human) behavior module", "go close to the target behavior module", and "talk with the target behavior module". The tree-structure allows a behavior designer to develop a situated behavior module easily by elaborating the behavior from the top of the tree to the leaves level.

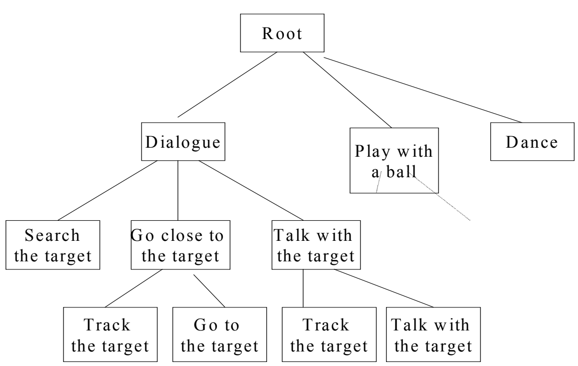

Fig. 11: An example of the tree-structure of situated behavior modules

Another advantage of the tree-structure is to share the behavior target. Assume that two different behaviors have to be executed in parallel. The first behavior is "dialogue behavior tree" whose target is a human to talk with, and the second behavior is "play with a ball behavior tree" whose target is a ball to play with. The target identifier of the face is commonly shared with the behavior sub-modules in the "dialogue behavior tree" and the target identifier of the ball is also commonly shared with the behavior sub-modules in the "play with a ball behavior tree". This mechanism enables to concurrently execution ofe multiple behaviors whose targets are different from each other.

Thus, the resource management using awith tree-structured architecture efficiently achieves the concurrent execution requirement.

3) State management of the situated behavior modules

In order to achieve the behavior interruption and resume requirement, we first consider that there should be at least three states for the behavior modules, which are "ready", "active", and "sleep" shown in a simple version in Fig. 12 (1). All behaviors are initialized as "ready" at the start. If the activation level of a behavior module is sufficiently high to warrant selection, then the behavior control system changes the state of the selected behavior module to "active". Now the action functions of the behavior modules with an "active" state are called so that the proper behaviors are executed. If the execution of the selected behavior module finishes properly, the system changes its state from "active" to "ready".

Now assume that the situation changes while some behavior modules are being executed, and a behavior module with a much higher activation level, which may be an emergency behavior, needs to be executed. Then, the system changes the state of the previously executing behavior module from "active" to "sleep" (not "ready"). After finishing the emergency behavior, the system changes the "sleep" behavior module back to "active" so that the behavior module can resume the action function from its suspended position.

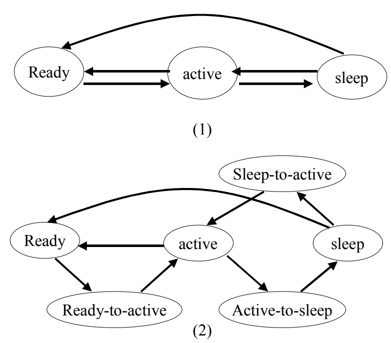

Fig. 12: State transition of a situated behavior module; (1) simple version, (2) revised version

For convenience, we further introduce the additional states, "ready-to-active", "ready-to-sleep", and "sleep-to-active" as shown in Fig. 12 (2). These additional states are used for natural interrupt handling of behaviors. Assume that SDR-4X is talking with person-A (with the dialogue behavior module), and then person-B calls to the SDR-4X from the left-hand-side of the robot. The behavior control system changes the state of the dialogue behavior module to "sleep", but it is more natural for the humanoid to first say, "just a moment please" to person-A. In general, we need an action sequence before a behavior module goes to sleep. In order to achieve this, the behavior control system changes the "active" behavior module to "active-to-sleep" state,

where this transition action sequence can be executed before the module goes to sleep.
Similarly, "sleep-to-active" state can be used to say "sorry for keeping you waiting" when SDR-4X finishes the task for person-B, and then returns to conversing with person-A.

VI. Implementation and Preliminary Experimental Results

We implemented the autonomous behavior control architecture described in the previous sections.
Fig. 13 shows the tree-structure of the situated behavior modules (right side of each photo), and the corresponding executing behaviors (left side of each photo). In the top left photo, the environment confirmation behavior module of the dancing behavior tree is activated. This behavior module checks the surrounding environment to see if it is a proper place to dance. In the tree-structure figure, there are some red-colored behavior modules. These are the behavior modules in the "active" state. shows the tree structure with the states corresponding to the top left photo in Fig. 13. We can observe that the dancing behavior module in the dancing behavior tree is activated.
In the bottom left photo, the rest behavior module is activated to take a rest for awhile after dancing. In the bottom right photo, the instinct/emotional expression behavior module is activated to display a happy behavior. Thus, behavior modules are selected and executed automatically so that spontaneous behavior generation is achieved.
In the photos, the human voice command in this tree-configuration is not activated, but we have already tested such a case as well.

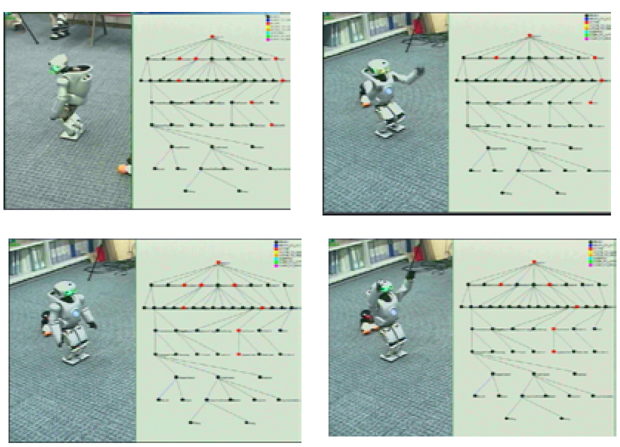

Fig. 13: Autonomous Behavior Generation 1

Fig. 14 shows the situation in which a user speaks to SDR-4X. In this situation, the user actively moves to SDR-4X's view range first, and start to speak to SDR-4X. In the top left photo, a user says, "hello" to SDR-4X, then the "greeting behavior module" is activated to say "hello" to the user. In the top right photo, the "language game behavior module" is activated and SDR-4X spontaneously talks with the user, by which SDR-4X asks some riddles of the human. In the bottom left photo, SDR-4X continues to talk with the user by asking for her birthday or other personal information. These language games are selected randomly by the parent "language game behavior module" when it is activated.

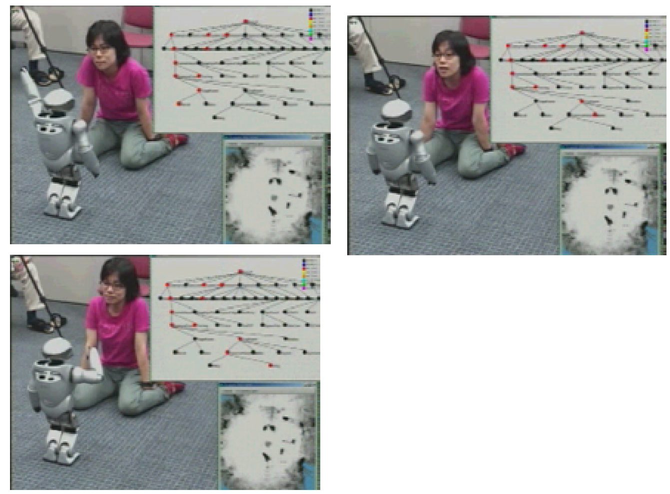

Fig. 14: Autonomous Behavior Generation 2

In addition to the above autonomous behavior control experiments, we evaluated the "behavior interrupt and resume" capability. This situation in fact often happens when we set the time-varying parameters rates as fast so that the internal variables quickly decrease to become out of their regulation ranges. This also occurs when SDR-4X falls down when an approaching behavior is activated, so then the recovery motion interrupts to suspend the approach behavior.
At the time of writing this paper, the experiments are not sufficiently organized and analyzed to discuss the problems of the architecture. We did not describe the experiments for concurrent execution capability of our architecture. We hope to have the opportunity to explain the experiments for the resource management method by the tree-structured architecture in the near future.

VII. Summary and Conclusions

We have described autonomous behavior control architecture for SDR-4X. The overall architecture is composed of the following parts: perception, memory, behavior control, and motion control. Compared to the software architecture of AIBO or SDR-3X, the memory part of the architecture is an essential new function. Short-term memory is necessary to remember the environment surrounding SDR-4X due to the limited view angle of the vision sensor. Long-term memory is necessary for SDR-4X to talk with humans with some level of intelligence. A dialogue using the long-term memory gives the perception of higher intelligence to a user than a dialogue without memory.

In this paper, we described the short-term memory in detail but not the long-term memory. We would like to describe the long-term memory in another paper focusing on dialogue issues.

In the last half of this paper, we described the autonomous behavior control architecture. The basic concept of behavior control is a situated behavior-based architecture, by which many situated behavior modules are implemented to cover the a wide range of real-world situations. Furthermore, we make a list of three basic requirements, which are, (1) multiple situated behavior modules can concurrently evaluate the situation, (2) multiple situated behavior modules can be concurrently executed with different mechanical resources, and (3) the situated behavior module can be suspended if an emergency behavior has to be executed, and then be resumed if the emergency behavior finish its task.

We introduced a monitor function for each behavior module to evaluate the situation, and an action function to achieve the task in the situation. We also introduce resource management for each behavior module so that the behavior control system can handle the concurrent execution requirement. Finally, we introduce state control of each situated behavior module so that the behavior control system can suspend the action function and then resume it as required.

Finally we implemented the autonomous behavior control architecture and show preliminary experimental result.

At the time of writing this paper, the experiments are not sufficient to explain the full implementation of the architecture. In the final version of this paper, we would like to add an explanation of the concurrent execution and behavior interrupt / resume capabilities.

Finally, for the future study, we would like to address some of the issues raised in this paper. The first one is regarding a personality for an autonomous robot. As we describe in section V, some parameters of ISM dynamics and the monitor function can control its behavioral tendencies, which we may call as a personality of an autonomous robot[13].

The second issue involves the Emotionally Grounded Symbol issue described in section III. This fundamental issue becomes practically important when we deal with symbol acquisition behavior, or behavior acquisition.

Thus, many challenges still remain in this field. We hope that the architecture described in this paper clearly defines these challenges so that we can realize a partner robot in the near future.

References

[1] Fujita, M. and Kageyama, K., “An Open Architecture for Robot Entertainment ”, Proc. of International Conference on Autonomous Agents, pp. 435-440, 1997.

[2] Fujita, M., and Kitano, H., “Development of An Quadruped Robot for Robot Entertainment”, Autonomous Robots, Vol. 5, pp.7—18, Kluwer Academic, 1998

[3] Ishida, T., Kuroki, Y., Yamaguchi, J., Fujita, M., Doi, T.T., “Motion Entertainment by a Small Humanoid Robot Based on OPEN-R ”, IROS, pp.1079-1086, 2001.

[4] Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., Doi, T.T., “A Small Biped Entertainment Robot ”, Proc. of the IEEE-RAS International Conference on Humanoid Robots, pp.181-186, 2001.

[5] Fujita M. et. al,, “A Small Humanoid Robot SDR-4X for Entertainment Applications”, International Conference on Advanced Intelligent Mechatronics, to appear, 2003

[6] Steels, L., “Perceptually Grounded Meaning Creation”, Proceedings of ICMAS-96, Kyoto 1996.

[7] Fujita M. et.al., “An Autonomous Robot that eats information via interaction with human and environment ”IEEE ROMAN-01,pp.383-389,2001.

[8] Fujita M., et. al., “Physically and Emotionally grounded symbol acquisition for autonomous robots ”, AAAI Fall Symposium, :Emotional and Intelligent II,pp.43-46,2001.

[9] Arkin, R. Fujita, M., Takagi, T., and Hasegawa, R., “Ethological Modeling and Architecture for an Entertainment Robot”, ICRA-2001

[10] Takayuki Kanda, Hiroshi Ishiguro, Michita Imai, Tetsuo Ono and Kenji Mase, An approach for developing interactive humanoid robots, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), 2002

[11] “The Talking Head Project” in http://www.csl.sony.fr/Research?Topics/Language/index.html

[12] Mataric, J. M., et. al. “Situated Robotics”, Encyclopedia of Cognitive Science, Nature Publisher Group, Macmillian Reference Ltd., 2002.

[13] Tappel R. (editor), Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agent, lecture note in Artificial Intelligence, Springer Verlag, 1997