Autonomous Robots, 5, 7–20 (1998)

Development of an Autonomous Quadruped Robot for Robot Entertainment *

MASAHIRO FUJITA
D21 Laboratory, Sony Corporation
mfujita@pdp.crl.sony.co.jp

HIROAKI KITANO
Sony Computer Science Laboratory Inc.
kitano@csl.sony.co.jp

;

Abstract. In this paper, we present **Robot Entertainment** as a new field of the entertainment industry using autonomous robots. For feasibility studies of Robot Entertainment, we have developed an autonomous quadruped robot, named **MUTANT**, as a pet-type robot. It has four legs, each of which has three degree-of-freedom, and a head which also has three degree-of-freedom. Micro camera, stereo microphone, touch sensors, and other sensor systems are coupled with newly developed behavior generation system, which has emotion module as its major components, and generates high complex and interactive behaviors. Agent architecture, real-world recognition technologies, software component technology, and some dedicated devices such as Micro Camera Unit, were developed and tested for this purpose. From the lessons learned from the development of MUTANT, we refined the design concept of MUTANT to derive requirements for a general architecture and a set of interfaces of robot systems for entertainment applications. Through these feasibility studies, we consider entertainment applications a significant target at this moment from both scientific and engineering points of view. [1]

Keywords: autonomous robot, quadruped robot, entertainment, agent architecture

1. Introduction

In this paper, we present **Robot Entertainment** as a new field for using autonomous robots in the entertainment industry. In order to demonstrate the feasibility of Robot Entertainment, we have developed a pet-type legged robot called **MUTANT** (Fig.1), which has four legs and head, each of which has three degrees-of-freedom and is equipped with on-board sensors such as a micro-camera, a stereo microphone, and touch sensors. It is also endowed with a behavior generation system consisting of an instinct/emotion module, a high-level cognition module, and reactive behavior subsystems. MUTANT interacts with people by tonal sounds, and exhibit a large variety of complex behavioral patterns. MUTANT was developed to investigate the feasibility of using robots as an entertainment tool.

* The original version of this paper is in Autonomous Robots 5, 7-18(1998), Kluwer Academic Publishers.
A premilinary version of this paper was presented at the First International Conference on Autonomous Agents in February 1997

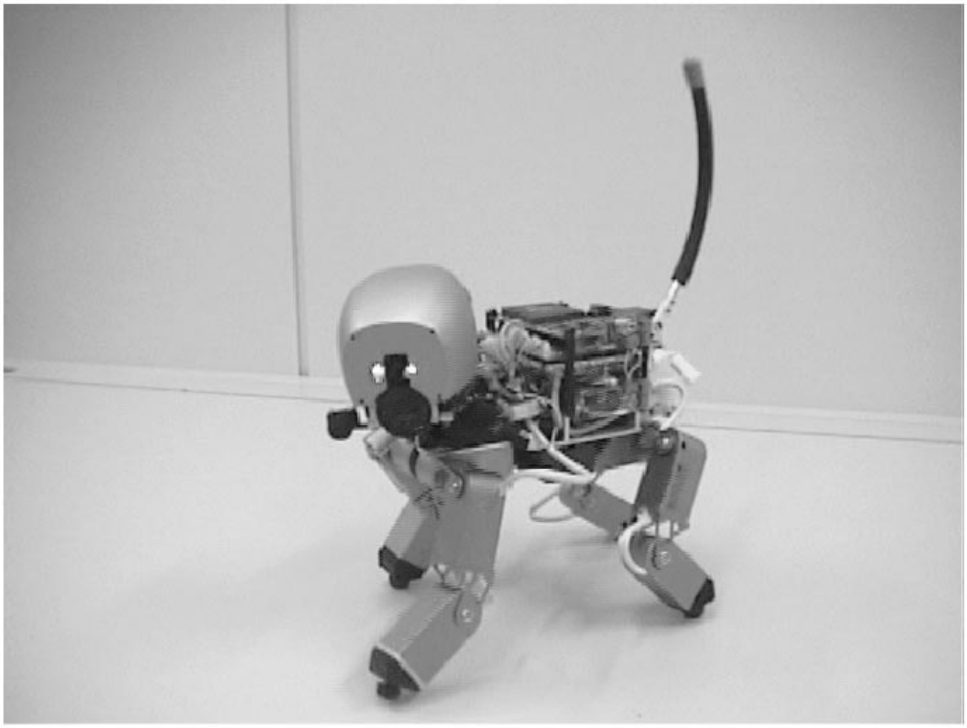

Fig. 1. MUTANT: an fully autonomous pet robot

We consider entertainment applications an important target at this stage of both scientific and industrial development. The three major reasons are:

Complete Agent: A robot for entertainment requires a completely autonomous physical agent. Instead of focusing on specific perceptual functions such as speech and vision, research on complete agents promotes and accelerates research activities involving the integration of subsystems.

Technology Level: A robot for entertainment can be effectively designed using various state-of-the-art technologies, such as speech recognition and vision, even though these technologies may not be mature enough for applications where they perform a critical function. While there exists special and difficult requirements in entertainment applications themselves, limited capabilities in the speech and vision systems may turn out to be an interesting and attractive feature for appropriately designed entertainment robots.

Emerging Industry: We believe that we will be able to create a completely new market in the near future by introducing this kind of robot products sharply focused on entertainment applications. We strongly believe that after the Gold Rush of the Internet and cyberspace, people will eagerly seek *real* objects to play with and touch. Robot Entertainment provides tangible physical agents and an unquestionable sense of *reality*.

It is expected that various kinds of entertainment applications will be developed for Robot Entertainment systems. The possibility of entertainment using such an autonomous robot is as follows:

Watching Motions of a Robot: Many people enjoy watching motions, gestures, and be-

haviors of animals in a zoo. Recently, computer controlled dinosaur robots have been very popular in theme parks. We consider that the ability to perform movements of a certain level of complexity is important for robots to really be entertaining.

Interacting with a Robot: We can also interact with a dog by gestures and voices. In addition to watching the motions of the pet animal, the interaction enhances the owner's enjoyment. We consider that interaction is also an important feature of entertainment robot.

Bringing up a Robot: Recently, simulation games, in which players bring up a virtual computer graphics agent, have popular in video game software. In the research field of Artificial Life (A-Life) and Artificial Intelligence (AI), researchers often discuss entertainment applications in this field (Maes, 1995, Kitano, 1994a, Kitano, 1995a).

Controlling a Robot: "Action" games are also popular in video game software. Many people enjoy controlling a car to compete for a time record, or a character to fight with another character controlled by another player. Robot soccer, for example, would be one of the most promising targets of applications. While RoboCup (Kitano, 1997b) focuses of research issues, consumers would certainly be interested in an easier way to play soccer with robots. Thus, controlling a robot will be one of the major application categories in Robot Entertainment.

Creating a Robot: We believe that the creation of a robot is itself entertaining. Recently, many robot contests have opened, where participants had to create robots under some constraints such as weight, cost, and so on. These participants enjoy creating robots implemented with their ideas, and enjoy observing the behavior of the robots they have designed. We believe that not only technically-trained people but also laymen can enjoy creating a robot, if we can provide a friendly development environment.

It is very likely that there will be more possibilities which we cannot imagine at this moment.

To confirm the above possibility of an entertainment robot, we developed an autonomous robot named **MUTANT**. In the following sections, we describe MUTANT, starting from its design concept, followed by the details of its implementation. Then, we discuss the issue of reusability of robot components, in terms of both hardware and software, for developing various kinds of applications, and for creating different robot configurations differing in shape, sensors, actuators, and so on.

2. Design Concept of MUTANT

To confirm the possibility of entertainment using autonomous robot, we developed a pet-type entertainment robot, named MUTANT. First, we defined the design concept of this prototype robot as follows:

Natural Human-Robot Interaction: Considering that interaction is an important feature of entertainment, robots should be capable of interacting with humans in a natural way without any special tools.

Real-World Complete Agent: In addition, robots should be capable of acting in an ordinary room environment, and avoid setting special conditions in terms of sound and illumination.

Complex Behavior: Considering that movements with a certain level of complexity are an important feature of entertainment, robots should be capable of acting with complex movements and complex behaviors.

Reusable Software: Considering that creating a robot is expected to be an important feature of entertainment, robotic software should be reusable in other robotic systems, in order to make easy the creation of other kinds of robots.

Standalone System with Extensibility: Robots should be a standalone pet-type robot, without any cables such as power lines, so that people regard the robot as a real pet. In addition, the robot should be easily extended to remote-operated system, which a user can control as a competition robot.

Our approach to accomplish the above design concept is the following:

- Use of a camera, a stereo-microphone, and a loudspeaker for natural interaction with humans.
- Tonal-language with lip-whistle for robust sound recognition in noisy environments.
- Complex motion with a quadruped mechanical configuration.
- Use of Object-Oriented Programming paradigm and software component technology.

3. Implementation of the Design Concept

3.1. Human-Robot Interaction

For a pet-type robot, one key to entertainment is how a robot interacts with the human, and vice versa. We choose to use a camera, a microphone, and a loudspeaker as main sensors and an effector, as shown in Fig.7, in order to allow natural interaction. In addition, we use a three-dimensional acceleration sensor and touch sensors which are used in low-level behaviors.

3.1.1. Micro-Camera-Unit. In order to make a robot small in size and weight and to reduce cost, we developed a Micro-Camera-Unit (MCU) using multi-chip-module technology (Ueda, 1996), as shown in Fig.2. This MCU includes a lens in the same package to obtain a single thin camera module. The size of the MCU is $23 \times 16 \times 4mm$, with pixels 362×492.

3.1.2. DC Geared-Motor. In addition to the MCU, in order to keep the robot small in size and weight, we developed a DC geared-motor for a link module of legs. It consists of a DC motor, five gears, a potentio-meter, and a motor driver chip (Fig.3). A motor case made of magnesium-alloy serves as the outer shell of a leg.

3.2. Real-World Complete Agent

For voice recognition, either Push-to-Talk techniques or a head-set microphone is used in many applications. However, it is better to interact with a robot without any special physical tools, such as a microphone with a switch. To achieve this goal, the following problems has to be resolved:

Noise: In an ordinary room or office environment, there are many noise sources (such as air-conditioning). In addition, a robot itself generates noise when it moves.

Voice Interference: There is also *voice interference* generated by other people in a room. For example, when we demonstrate our robot, people often talk to each other around the robot.

In general, the ability to distinguish the desired sound source in the noisy environment is referred to as the *cocktail party effect.* A human is able to distinguish a target voice from others voices and noise in a cocktail party. Pet-robots also should be able to exhibit a similar capability, possibly with specialized communication methods.

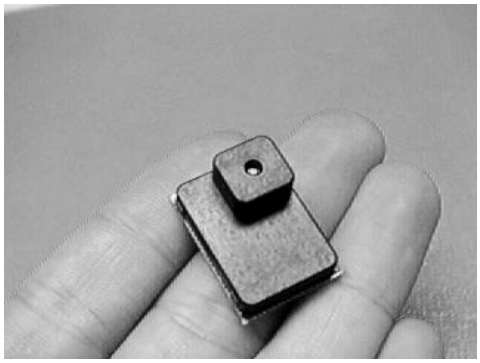

Fig. 2. Micro Camera Unit

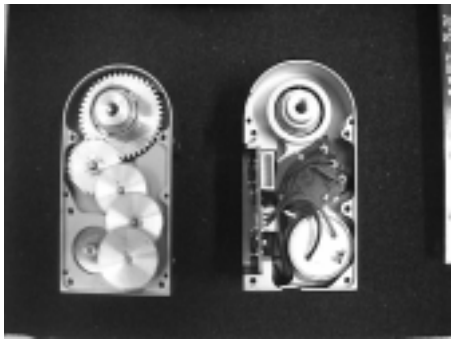

Fig. 3. DC Geared-Motor

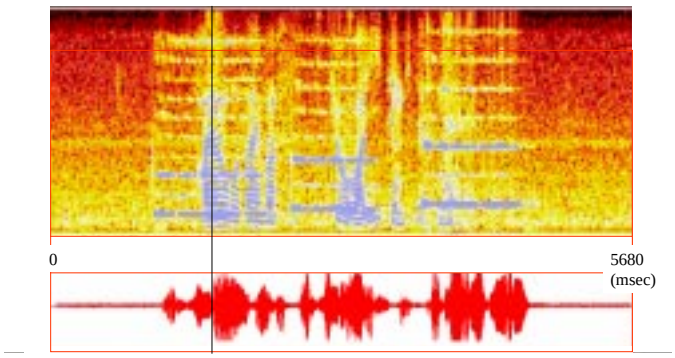

Fig. 4. Time-Frequency Characteristics of Tones with Voices: the sequence of the tones consists of "fa (349.23Hz)", "ra (440.00Hz), and "do (523.25Hz)". While the sequence of the voice is "ohayou gozaimasu. kyou ha ii tenki desune" of a Japanese male speech. At the slice point by the vertical line (the 100th frame), the tone frequency is 349.23Hz.

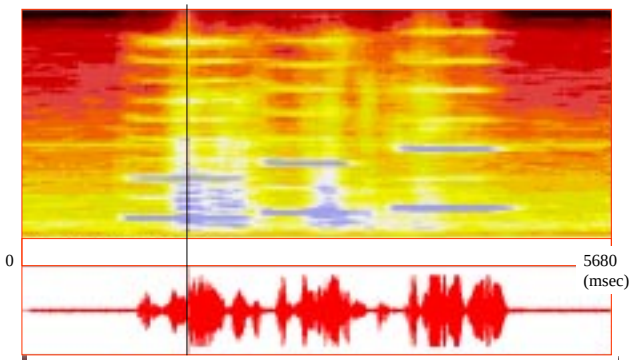

Fig. 5. Filtered Time-Frequency Characteristics of Tones with Voices

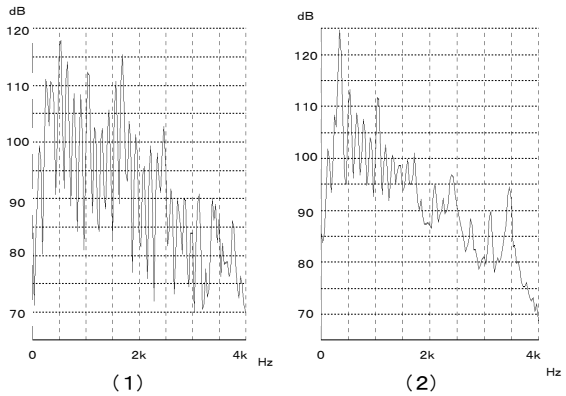

Fig. 6. Spectrum of the 100th frame: (1) Original Data, (2) Filtered Data "showing the recovered tone at 349.23Hz and its harmonics frequency components"

ponents for a human to interact with a robot is to use the musical scale, "A-B-C", as the basic components of communication.

This approach can also solve the second problem, voice-interference in a room. In general, the pitch component of a spoken voice in natural conversation does not keep the same frequency, but varies (Fujisaki, 1984). On the other hand, a musical scale signal can keep a constant frequency. Fig.4 shows a time-frequency graph of a mixture of natural conversation and musical tones by a synthesizer, and Fig.5 shows a time-frequency graph of filtered version. We use a "moving-average" filter with a 400msec averaging window.

Fig.6 shows the spectrums of both the original signal and the filtered signal. In the left-hand graph, the target tone pitch frequency is suppressed by the voice interference; however, in the right-hand graph, the target pitch frequency is recovered by filtering the varying voice pitches.

Since a human can generate such a signal with a lip-whistle or his/her voice, it is possible to interact with a robot without using any special tools in a natural environment.

3.3. Complex Behavior

3.3.1. Quadruped Mechanical Configuration. Most of the intelligent autonomous robots are implemented in wheel-based mechanical configurations. Using the Subsumption Architecture, Brooks(Brooks89a) implemented a six-legged walking robot which presented complex behaviors while dynamically interacting with a real-world environment. At least on video, the impact of presenting complex motions of the six-legged robot is bigger than that of wheel-based robot.

We believe that the capability of representation and communication using gesture and motion is very important in entertainment applications. Therefore, the mechanical configuration we chose for MUTANT is that of a quadruped, as shown in Fig.7. The merits of the quadruped configuration are, (1) walking control of a quadruped is easier than that of a biped robot, and (2) when in a posture of sitting, two "hands" are free to move, and allow the display of emotions, or to communicate with a human through hand motions. Thus, because each leg or hand has to be used for various purposes besides walking, we assign three degrees-of-freedom (DoF) for each leg/hand. In addition, we add a tail and three DoF for neck/head so that MUTANT has enough representation and communication capabilities using motions.

3.3.2. Agent Architecture. Another feature in MUTANT for making complex behaviors is to have various kinds of behavior and motion control subsystems, which can at times compete or cooperate. Through competitions and cooperations, complex motions and actions emerge naturally.

There are three dimensions in the competition and cooperation of behavior controls:

Reaction/Deliberation: The question of how to coordinate reactive behavior, as in the subsumption architecture, and deliberation-based behaviors driven by high-level cognitive processing and emotional status is one of the central issues in agent architecture. Looking from the aspect of response time from sensor inputs to action outputs, they can be distinguished as (1) reaction with a fast response time, and (2) deliberation with a slow response time. In addition, these two aspects of control are also subject to competition and cooperation as described below.

Instinct/Emotion/High Level Cognition: Within deliberative actions, there are three levels of underlying control subsystems: instinct (or drive), emotion, and high-level cognition. An example of behaviors motivated by instinct (or drive) is an action of moving toward a battery station to *eat* (electricity) when the battery level is low. This class of behavior is triggered, in general, by body needs. An example of behaviors motivated by emotion is the getting angry. Both body needs and situation where the agent is in affect emotional status and their expression. A behavior motivated by high level cognition is, for example, to communicate with a human using motion and voice. This requires some level of planning and reasoning, but is often affected by emotional status.

Hand/Neck/Body: Some of the reactive behaviors can be embedded in each body components. Each part of a robot, such as a right hand, is able to move autonomously with its

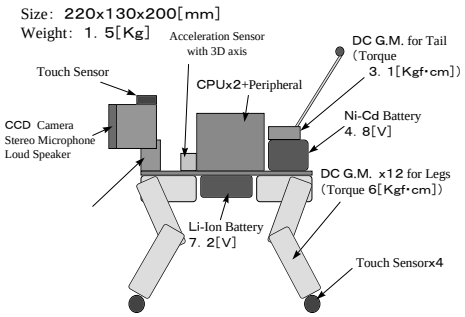

Fig. 7. Mechanical Configuration of MUTANT

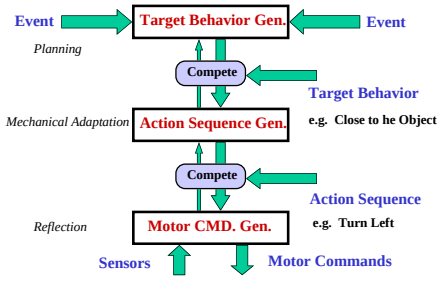

Fig. 8. Layered Architecture for Reaction and Deliberation

own control subsystem. For examples, the neck moves to track an object, a hand moves to touch other objects, and the entire robot moves to come close to something. The combination of these motions/actions generated by the parts makes it possible for the robot to exhibit very complex behaviors. In order to generate coordinated reactive behaviors, each body component needs to be cooperative. However, there are cases where two actions are conflicting, and therefore require the selection of one action over the other.

These subsystems are integrated by the following methods;

Layered Agent Architecture: We design the agent architecture of MUTANT as a combination of reactive processes and deliberation processes. As shown in Fig. 8, sensory inputs can be processed and a motor command can be sent back immediately by the motor command generator. Higher level processing can be programmed using action sequence generators and target behavior generators.

Instinct/Emotion Module: We designed an instinct/emotion module in the agent architecture so that MUTANT can have a capacity for spontaneous and emotional behaviors. Since this implementation was just a trial for MUTANT to have the capability of complex behaviors, at this moment, we only assign three instinct/emotional states: novelty-boredom, fatigue-activation, and happiness-anger, to MUTANT. Some of the sensors provide input to this module to vary the states which generate some emotional target behaviors such as seeking, sleeping, or getting angry. Although this model is simple, it works to increase the complexity of MUTANT's behaviors. We are currently implementing full-scale emotional/instinct module with rich emotional and body status parameters.

Tree-Structured Architecture: In order to achieve independence of each part of the robot, such as a hand, or a neck, we use the tree structure topology of robot parts configuration, as s hown in Fig.10. In addition, we assign the layered agent architecture to every node in the tree structure. This tree structured agent architecture enables the independence of each part. Competitions and cooperations of these node motivations are carried out by using communication through the tree branches. Currently, for competition and cooperation, MUTANT uses the simplest rule, which is that the control/motivation of the upper part has higher priority. The current implementation of the Tree-Structure has only two nodes, a head and a body.

Fig.11 shows the entire agent architecture of MUTANT, including *Layered Agent Architecture* for a body and a head, *Instinct/Emotion Model*, and *Tree-Structured* Architecture for autonomy of a body and a head.

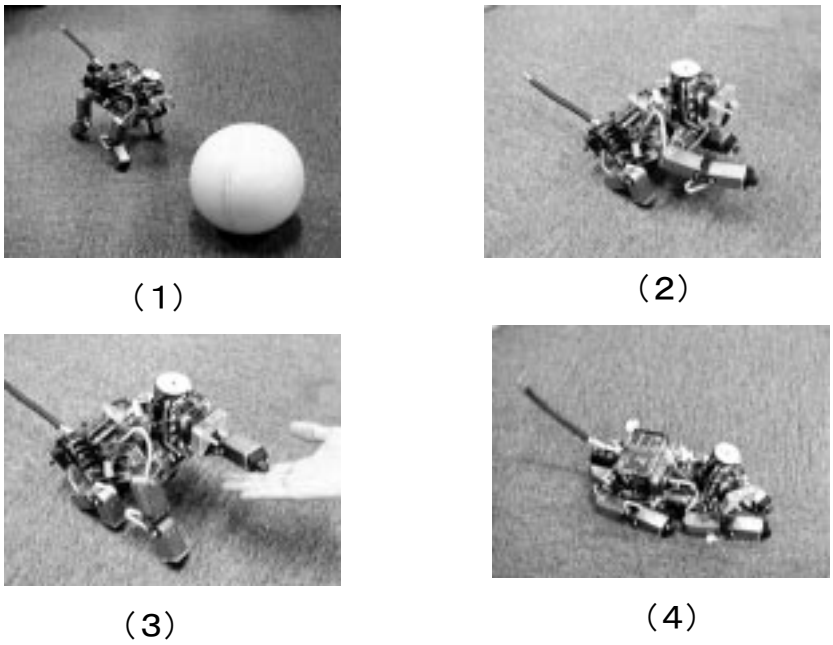

Fig. 9. MUTANT: (1) tracking a yellow ball, (2) karate action with anger, (3) shake hands, and (4) sleeping

Fig.9 shows the complex behaviors displayed by MUTANT with its quadruped mechanical configuration and the agent architectures. The "tracking a yellow ball" behavior is carried out reactive behavior, the "karate" behavior is motivated by the instinct/emotional state of anger, the "shake hands" behavior is motivated by color detection of skin color and obstacle distance estimation, and the "sleeping" behavior is motivated by the instinct/emotional state of fatigue.

3.4. Reusable Software

One of the major results of recent software engineering is object-oriented programming (OOP) technology(Meyer, 1988), which aims at reusable software production. We use OOP technology for software development for MUTANT.

In addition, we use the Observer Pattern in Design Pattern technology (Gamma, 1995) in order to make each object independent from others. In general, according to the OOP, we define the interface methods for each object, and when using this object, designers for other objects use the defined interface methods. For instance, Object-A processes some data and gets a result, and Object-B uses the interface of Object-A to get this result. In this case, Object-B must know both the interface to get the result and Object-A itself. Thus, Object-B is not independent from Object-A, and this reduces the reusability of Object-B. Observer Pattern solves this problem by defining a subject and observers. In the above case, Object-A is subject to output the data and Object-B is the observer of the data. Each object must know only the type of the data. An observer is dynamically attached to the subject to identify which objects to send the result data to. Thus, the designer of each object does not need to know the

Table 1. Features of MUTANT

Size and weight	
Type	quadruped legged robot
Size	$220 \times 130 \times 200[mm]$
Weight	$1.5[Kg]$ (including batteries)
Dof	sixteen
Actuators	
Type	DC geared-motor with a potentio-meter
Torque	$6[Kgf \cdot cm]$ for legs
	$3.1[Kgf \cdot cm]$ for a neck and a tail
Sensors, effectors, and batteries	
Sensors	CCD camera, stereo microphone
	3D acceleration sensor, touch sensors
Effector	a loud-speaker
Battery	Li-ion (7.2V) for electric circuits
	Ni-Cd (4.8) for motor drivers
CPU	
Type	IDT R3052 or R3071 $\times 2$
Clock	about $30[MHz]$
RAM	$8[MBytes]$
Flash ROM	$2[MByes]$

other objects. This independence results in software reusability and efficient programming development.

3.5. Standalone System with Extensibility

The pet-type robot is a standalone robot, whose battery, CPU board, and every component are on board. Fig.12 shows another application, with a remote-operated System, for soccer game playing. We add a wireless receiver to MUTANT, and modify the software in the top layer with command interpreter for the wireless command signals.

3.6. Summary of the Features

The mechanical and electrical features of MUTANT are summarized in Table.1.

4. Discussion

In order to develop various kinds of Robot Entertainment systems, we believe that reusability of subsystems, in terms of both software and hardware, is very important. This leads us to define **OPENR** [2], a standard architecture and a set of interfaces for Robot Entertainment Systems.

We argue that four types of scalability are essential for such a standard. These are (1) size scalability, (2) category scalability, (3) time scalability, and (4) user expertise scalability.

Size Scalability (Extensibility): Size scalability means that robotic systems can be scaled up in terms of their number of components

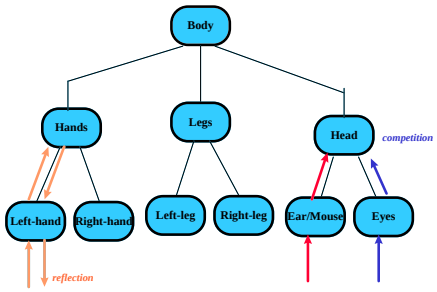

Fig. 10. Tree-Structured Architecture for Sub-System Autonomy

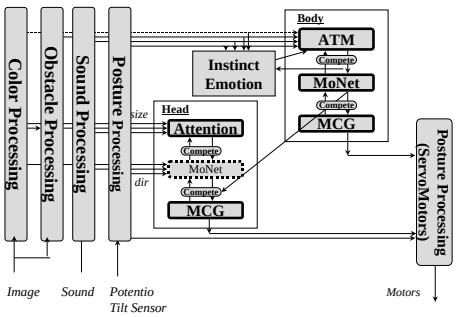

Fig. 11. Entire Agent Architecture of MUTANT

and number of robots involved in one system. OPENR must be extensible for various system configurations. For example, in a minimum configuration, a robot may only be composed of a few components, and perform a set of behaviors as a complete agent, without any external system. Such a robot can be scaled up in terms of the number of components by adding physical components. It should also be possible to scale up a system by having such robots as sub-systems of large robotic systems. Each robot should be able to work as a part of a larger system, as a remote-operated system.

Category Scalability (Flexibility): Category scalability ensures that various kinds of robots can be designed based on the OPENR standard. For example, two very different styles of robots, such as a wheel-based robot and a quadruped robot, should be able to be described by the OPENR standard. These robots may have various sensors, such as cameras, infra-red sensors, and touch sensors as well as motor controllers.

Time Scalability (Upgradability): Time scalability guarantees that users can always use up-to-date components, by merely replacing old modules. OPENR must be able to evolve together with the progress of hardware and software technologies. Thus, it must maintain a modular organization so that each component can be replaced with up-to-date modules.

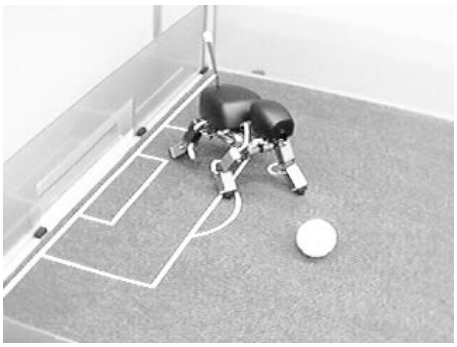

Fig. 12. Remote-Operated Soccer Game Player

User Expertise Scalability (Friendly Development Environment): OPENR must provide a development environment, both for professional developers and for end-users who do not have technical knowledge. End-users may develop or compose their own programs using the development environment. Thus, it should be scalable in terms of the level of expertise that designers of the robot have.

Our approach to meet these requirements are itemized as follows:

- Generic Reference Model, for developing various kinds of robot architectures, with a friendly development environment,
- Configurable Physical Component for reusable robot hardware.
- Object-oriented programming for reusable software components,
- Layering technique for upgradability.

Details of the approach are described in (Fujita and Kageyama, 1997a). We briefly overview OPENR in Appendix.A.1.

5. Related Works

In Robot Entertainment, the shape of each robot should be freely customizable for each application. Most existing robot platforms are not able to customize their shapes, such as YAMABICO(Iida, 1991). Our idea of Configurable Physical Component solves this problem so that designers are able to design various shapes of robots.

Inaba proposed a research and development approach named "Remote-Brain Approach" for autonomous robot (Inaba, 1993). His group has implemented various shapes of robots using this approach, in which computers are not on the robot body, but are remote-hosts with wireless transmitters. Therefore, they can use significant computer power while keeping the size of robot body small. Our architecture can be a standalone robot as well as a remote-controlled robot with wireless extension module using the limited bandwidth of wireless communication channel. The capability of standalone robot without host computers is important when motor commands from host computers cannot be received because of bad wireless

channel condition. The humanoid-type robots developed in his laboratory efficiently show the complex motions and behaviors, which we listed as one of the important features for entertainment using autonomous robots.

Robot Entertainment requires satisfy that motions and behaviors of robots be sufficiently complex or unexpected so that people keep an interest in watching or taking care of it. Complex motions and behaviors may be realized using the Subsumption Architecture (Brooks, 1986) or in general Behavior-Based Approach, in which the complexity of motions and behaviors depends on the complexity of environments.

In addition, Robot Entertainment requires that behaviors of the robot should be able to be designed according to a designer's intention. For an autonomous pet-type robot, for example, a designer may want to design a scenario of a robot life, and its personality. For a music lover dancing robot, a designer may want to design its motions. These motions and behaviors are usually expectable and are realized using motion databases or classical AI technology.

The two requirements, complexity and design capability of motions and behaviors, are similar to the requirements, reactive and deliberate behaviors in many Agent Architectures (Hayes-Roth, 1995, Noreils, 1993, Bonasso95). For example, Noreils's architecture (Noreils, 1993) has three layers (levels); functional level, control level and planning level. In the functional model there are many reactive behavior modules which are controlled by the controlled level. In the planning level, deliberate planning is examined. Thus, reactive and deliberate behaviors are both achieved with this architecture.

This technique, the combination of reaction and deliberation, is used in MUTANT as described in the previous section. In our robot, however, planning is more primitive than that of the above Agent Architecture.

Finally, Maes (Maes, 1995) points out that the entertainment area may become much more important for agent research. She introduces ALIVE, a virtual environment allowing wireless full-body interaction between a human and a virtual world which is inhabited by animated autonomous agents. There are some other examples of entertainment agents, such as (Tosa, 1993). Although these agents inhabit virtual worlds, many research topics are very similar to those of Robot Entertainment.

6. Conclusion

We presented Robot Entertainment as a new field of research and industry for real-world AI and robotics. To confirm the possibility of entertainment using autonomous robots, we developed an autonomous robot, named MUTANT, with a design concept that includes (1) natural human-robot interaction, (2) the realization of a real-world complete agent, (3) complex behaviors, (4) software reusability, and (5) extensibility. To realize the design concept, we developed a micro camera unit (MCU) using a multi-chip-module technology, and a DC geared-motor to make MUTANT small in size and weight. The features are summarized in Table.1 (also see Fig.7).

We also tested the tonal-language with musical sound, the agent architecture, and object-oriented programming in the quadruped robot for a pet-type application. Through the development of MUTANT, we are refining our design concept, stressing the importance of the reusability of subsystems or components for establishing Robot Entertainment as an emergent industry. We have discussed a standard architecture and a set of interfaces, and the requirements which such a standard must satisfy.

We believe that Robot Entertainment is significant at this moment, from both the scientific and the engineering points of view, and that we will be able to create a completely new market in the near future.

Acknowledgements

The authors would like to thank to Dr. Stephane Zrehen at D21 Laboratory, Sony Corporation for his English revising of this paper.

Appendix
Overview of OPENR

A.1. Overview of OPENR

A.1.1. Requirements

As we discussed in secion.4, the four types of scalability are essential for OPENR.

- Size Scalability (Extensibility)
- Category Scalability (Flexibility)
- Time Scalability (Upgradability)
- User Expertise Scalability (Friendly Development Environment)

A.1.2. Strategy

Our tactics to meet these requirements are itemized as follows:

- Generic Reference Model, for developing various kinds of robot architectures, with friendly development environment,
- Configurable Physical Component for reusable robot hardware.
- Object-oriented programming for reusable software components,
- Layering technique for upgradability.

A.1.2.1. Generic Reference Model. We define a generic system functional reference model (GSFRM) composed of Basic System, Extension System and Development System, as shown in Fig.A.1. By defining GSFRM, we are able to construct various kinds of robot systems, such as a standalone robot system, a robot connected with a host computer, a robot with network link, and multi-agent robot system, as in Fig.A.2.

A.1.2.2. Configurable Physical Component. OPENR defines common interfaces for all robot components in order to achieve flexible and extensible robot configurations. The physical connection between the robot components is done by a serial bus. For example, using the common components, we can create both a quadruped-type robot and a wheel-based type robot, as shown in Fig.A.3. In addition, every CPC has non-volatile memory with (1) functional properties, such as an actuator and a camera, and (2) physical properties such as the three dimensional size of a part.

With this information, it is possible for the host controller of the serial bus to know what kinds of components are part of the robot. Furthermore, as shown in Fig.A.4. in managing the tree connected structure, the host controller can get information on how all the components are connected to each other. This mechanism makes it possible for users to create their own robot configurations with favorite components.

A.1.2.3. Software Component. The idea of decomposing the system into components is extended to the software architecture of a robot. The key idea is to employ an object-oriented operating system, Aperios(Yokote, 1992), and to keep each software object as independent as possible. The Observer Pattern technique is useful for this purpose as we verified in the MUTANT software architecture.

A.1.2.4. Layering. The purpose of layering is flexibility and easy development of hardware and software components. OPENR divides each functional element into three layers, Hardware Abstraction Layer (HAL), System Service Layer (SSL), and Application Layer (APL).

Hardware abstraction software in the HAL makes it possible for users to reuse device driver software by providing a standard interface defined by OPENR. The SSL provides basic services to users. The Virtual Robot described above is one such example. The APL provides a more convenient service than that of the SSL. The Designed Robot is one of such example.

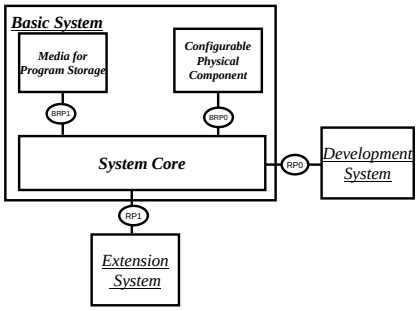

Fig. A.1. Generic System Functional Reference Model and Basic System Functional Reference Model

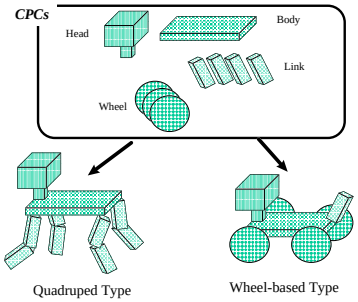

Fig. A.3. Various styles of robot with CPC

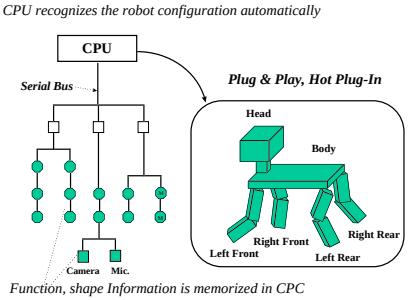

Fig. A.4. Configurable Physical Component

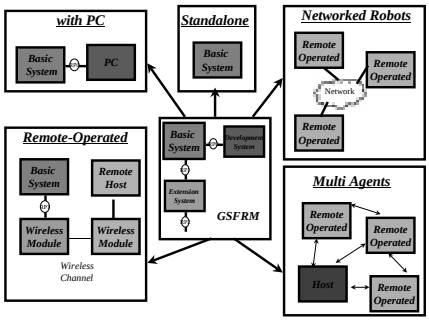

Fig. A.2. Various Architectures derived from GSFRM

Notes

1. Some parts of the contents in this paper are the same as in the paper in the 1st international conference on Autonomous Agents 1997(Fujita and Kageyama, 1997a). In this paper, we emphasize **Robot Entertainment** itself rather than the **OPENR** which was the main issue in Autonomous Agents97. We emphasize a design concept of a prototype quadruped robot, and we describe how we realize the concept, including agent architecture issue.
2. **OPENR** is a trade mark of Sony Corporation

References

Albus, James S., 1996, The Engineering of Mind, In *Proceedings of the fourth international conference on simulation od adaptive behavior (SAB96, From animals to animats 4)*, pp.23–32.

Bonasso, R. P., Kortenkamp, D., and Miller, D., 1995, Experiences with an Architecture for Intelligent, Reactive Agents, In *1995 IJCAI Workshop on Agent Theories, Architectures, and Languages*, August.

Brooks, R. A., 1986, A Robust Layered COntrol System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, RA-2(1), March, pp.14–23.

Brooks, R. A., 1989, A Robot that Walks: Emergent Behavior from Carefully Evolved Network, *Neural COmputation*, 1, pp.253–262.

Firby, R. J., 1994, Task Networks for Controlling Continuous Processes, In *Proceedings on the Second International Conference on AI Planing Systems*, June.

Fujisaki, H. and HIrose, K, 1984, Analysis of Voice Funcdamental frequency contours for declarative sentenses of Japanease, J. Acoust. Soc. Jpn., 5(4), pp.233-242.

Fujita, M. and Kageyama, K., 1997, An Open Architecture for Robot Entertainment, In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Ray, pp.234–239.

Fujita, M., Kitano, H. and Kageyama, K., 1997, A Legged Robot for RoboCup based on OPENR, In *The First International Workshop on RoboCup in conjunction with IJCAI-97*, Nagoya, pp.69–74.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., 1995, *Design Pattern: Reusable Object-Oriented Software*, Addison-Wesley.

Hayes-Roth, B., 1995, An architecture for adaptive intelligent systems, *Artificial Intelligence*, 72(1-2), pp.329–365.

Iida, S. and Yuta, S., 1991, Vechicle command system and trajectory control for autonomous mobile robots, In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems 91*, pp.212–217.

Inaba, M., 1993, "Remote-Brained Robotics: Interfacing AI with Real World Behaviors, In *Proceedings of the 6th International Symposium on Robotics Research (ISRR6)*, pp.335–344.

Kitano, H. (ed), 1994, *Proceedings of AAAI Workshop on Entertainment and AI/Alife*, Seattle.

Kitano, H. (ed), 1995, *Proceedings of AAAI Workshop on Entertainment and AI/Alife*, Montreal.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E., 1997, RoboCup: The Robot World Cup Initiative, In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, pp.340–347.

Kitano, H., et al, 1997, RoboCup: A Challenge Problem for AI, *AI Magazine*, Spring, pp.73–85.

Noreils, F. R., 1993, Toward a Robot Architecture Integrating Cooperation between Mobile Robots: Application to Indoor Environment, *The International Journal of Robotics Reseach*, 12(1), February, pp.79–98.

Maes, P., 1995, Artificial Life meets Entertainment: Lifelike Autonomus Agents, *Communication of the ACM: Special Issue on New Horizons of Commercial and Industrial AI*, pp.108–114.

Meyer, B., 1988, *Object-Oriented Software Construction*, Prentice Hall.

Soukup, Jiri, 1994, *TAMING C++: Pattern Classes and Persistence for Large Projects*, Addison-Wesle.

Tosa, N., 1993, Neurobaby, In *SIGGRAPH-93 Visual Proceedings, Tomorrow's Realities*, ACM, 1993, pp.212–213.

Ueda, K. and Takagi, Y., 1996, Development of Micro Camera Module, In *Proceedings of the 6th Sony Research Forum*, pp.114–119.

Yokote, Y., 1992, The Apertos Reflective Operating System: The Concept and Its Implementation, In *Proceeding of the 1992 International Conference of Object-Oriented Programing, System, Languages, and Applications.*

Masahiro Fujita is a senior reseach scientist at D21 Laborotary, Sony Corporation. He received a B.A. degree in Electoronics and Communication from the Waseda University, Tokyo, in 1981, and a M.S. degree in Electrical Engineering from the University of California, Irvine, in 1989. His research interests include Modular Robotics, Neural Networks, Visual and Sound Early-Perception, Face Recognition, Attentional Mechanism, and Emotional Model.

Hiroaki Kitano is a senior resaecher at Sony Computer Science Laboratory. He received a B.A. in physics from the International Christian University, Tokyo, and a Ph.D. in computer science from Kyoto University. He was a visiting researcher at Carnegie Mellon University during 1989-1993. Kitano received the Computers and Thought Award from the International Joint Conference on Artificial Intelligence in 1993. Kitano is a founder of RoboCup, and the Chair of The RoboCup Federaion. His reseach interests include RoboCup, computational molecular biology, engineering use of the mophogenesis process, and evolutionary systems.