A Legged Robot for RoboCup Based on "OPENR"

Masahiro Fujita[1], Hiroaki Kitano[2] and Koji Kageyama[1]

[1] D21 Laboratory, Sony Corporation,
6-7-35, Kitashinagawa,
Shinagawa-ku, Tokyo, 141 JAPAN
[2] Sony Computer Science Laboratory Inc.
Takanawa Muse Building, 3-14-13 Higashi-gotanda,
Shinagawa-ku, Tokyo, 141 JAPAN

Abstract. We propose a physical robot platform for RoboCup based on **OPENR**, which is proposed as a standard architecture for **Robot Entertainment**, a new entertainment field devoted to autonomous robots. The platform will be provided as a complete autonomous agent so that researchers are able to start working within their research fields. It is possible to rewrite all of, or parts of, the software of the RoboCup player. The configuration of the platform is quadruped type with a color camera, a stereo microphone, a loudspeaker, and a tilt sensor. In order to reduce the computational load, we propose that the walls of the field and the player itself are painted in different colors so that it is easy to identify the position of essential objects.

1 Introduction

In this paper, we propose a physical robot platform for RoboCup. RoboCup[3] was proposed as a new standard problem for A.I. and robotics research. RoboCup aims at providing a standard task for research on fast-moving multiple robots, which collaborate to solve dynamic problems. Although building a real robot for RoboCup is an important aspect of robotics, it is a money and time-consuming task for academic researchers. Thus, researchers are eager for a physical robot platform on which it is easy to build their academic research.

Previously, we proposed **OPENR**[3] as a standard architecture and interfaces for **Robot Entertainment**, which is a new entertainment field using autonomous robots[2]. OPENR aims at defining a standard whereby various physical and software components which meet the OPENR standard can be assembled with no further modifications, so that various robots can be created easily.

One of the purposes of proposing OPENR is that we wish to further promote research in intelligent robotics by providing off-the-shelf components and basic robot systems. These robots should be highly reliable and flexible, so that researchers can concentrate on the aspect of intelligence, rather than spending a substantial proportion of their time on hardware troubleshootings.

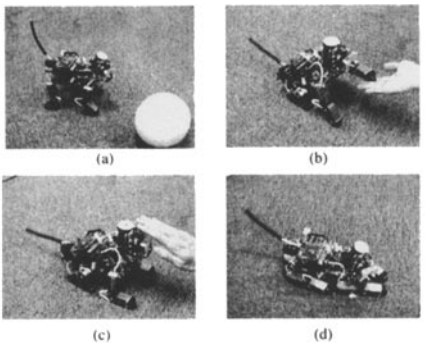

Fig. 1. MUTANT: Fully autonomous pet-type robot

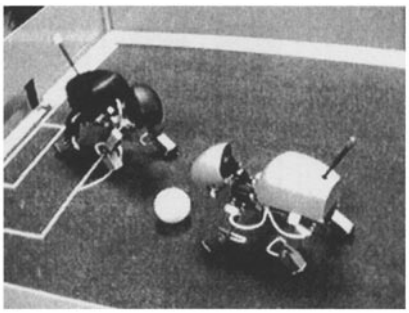

Fig. 2. Remote-operated soccer game robot

For the feasibility study of OPENR, we developed an autonomous quadruped pet-type robot named MUTANT[2] (Fig.1). This robot did not implement all ideas of OPENR, but it allowed us to test some of the software components and the agent architecture proposed in OPENR. By adding an extension module, we also implemented and tested a remote-operated robot system for soccer game (Fig.2). Using these robots, software and application feasibility studies were carried out.

As the next step, we have been developing new dedicated LSIs for the next generation prototype robot which meets the OPENR standard completely. We propose a platform for a RoboCup player based on this new prototype robot. In addition, we also propose a game environment such as a field and a ball, and some algorithms suitable for this platform.

[3] **OPENR** is a trade mark of Sony Corporation.

2 OPENR

2.1 Requirements

In order for OPENR to be accepted as an open standard for entertainment robot, there are several requirements which OPENR should satisfy. We argue that four types of scalabilities are essential for such a standard. These are (1) size scalability, (2) category scalability, (3) time scalability, and (4) user expertise scalability.

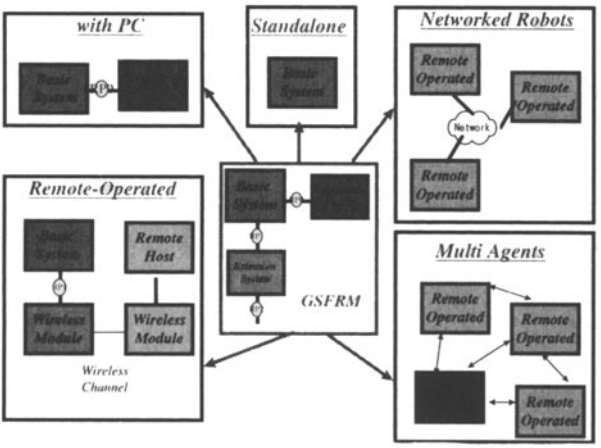

Fig. 3. Generic System Functional Reference Model and Examples of Derived System Architectures

Size Scalability (Extensibility): OPENR must be extensible for various system configuration. For examples, in a minimum configuration, a robot may be composed of only few components, and perform a set of behaviors as a complete agent. This robot can be scaled up by adding additional physical components. It should also be possible to scale up such a system by having such robots as sub-systems of large robot systems.

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-legged robot, should be able to described by the OPENR standard. These robots may have various sensors, such as cameras, infra-red sensors, and touch sensors and motor controllers.

Time Scalability (Upgradability): 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.

User Expertise Scalability (Friendly Development Environment): OPENR must provide a development environments, 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 is scalable in terms of the level of expertise that designers of the robot have.

2.2 OPENR Strategy

Our strategy to meet these requirements consists of the following:

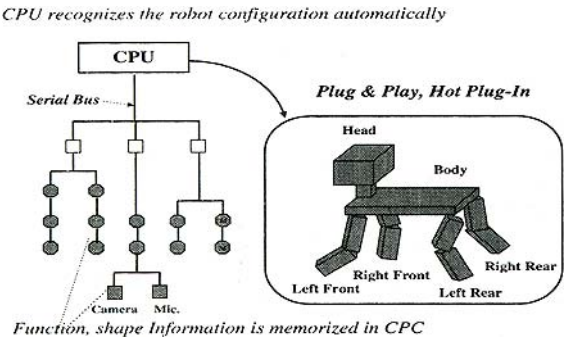

Fig. 4. Configurable Physical Component

Generic Reference Model: To meet the requirements of Extensibility and Friendly Development Environment, we define a generic system functional reference model (GSFRM) composed of Basic System, Extension System and Development System. By defining GSFRM, we are able to construct various kinds of robot systems with extensibility and development environments, as shown in Fig.3.

Configurable Physical Component: To meet the requirements of Flexibility and Extensibility, we devise a new idea of Configurable Physical Component (CPC). The physical connection between the robot components is done by a serial bus. In addition every CPC has non-volatile memory with (1) functional properties, such as an actuator and a two dimensional image sensor, and (2) physical properties, which help solve the dynamics of the robot consisting of these CPCs.

Object-Oriented Programming: To meet the requirements of Up-gradability and Flexibility, we employ an object-oriented OS, **Aperios** [6], which supports the Object-Oriented Programming (OOP) paradigm from the system

level with several types of message passing among objects. In addition, Aperios is capable of customizing APIs by system designers.

Layering: To meet the requirements of Up-gradability and Friendly Development Environment, we utilize the layering technique which is often used for multi-vendor open architecture. OPENR divides each functional element into three layers, Hardware Abstraction Layer (HAL), System Service Layer (SSL), and Application Layer (APL), as shown in Fig.5

In order to achieve the software component concept, an Agent Architecture for entertainment application was studied. The details is described in [2].

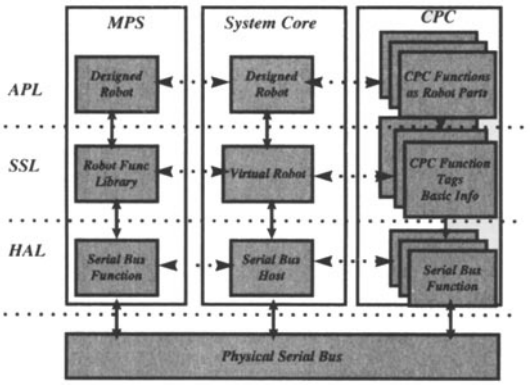

Fig. 5. Layering for Basic System

3 Proposed RoboCup System

3.1 Basic Behavior of RoboCup Player

Although RoboCup includes multi-agent co-operation and team play problems, at the primary stage, let us consider an individual player's behavior. The typical action sequence in this situation is (1) to find the ball, (2) to get close to the ball, (3) to kick or to dribble the ball toward the opponent goal, (4) to shoot to get a goal.

Thus, in this simple case the RoboCup player must somehow handle the position information of the following objects:

1. the ball,
2. the opponent goal,
3. the player itself.

3.2 Proposed RoboCup System

In general, it is difficult for a standalone robot system to perform these tasks in real time in a real world environment because of its limited computational power. The remote operated robot system depicted in Fig.3 can solve the computational power problem; however, in general, much computational power is necessary for image processing tasks. This implies that it is necessary for each robot to be equipped with a video transmitter. This is sometimes difficult for regulation reason.

Another solution to this problem is to set up a camera overlooking the entire field, and to distribute the image signal to all host computers[1]. In this RoboCup architecture, each robot can use the huge computer power of the host computer, or a special engine such as image processing hardware. However, the image information taken by the overlooking camera is not the image taken from each robot's viewpoint.

We consider that technologies to process the image from each robot viewpoint without any global information will become very important in Robot Entertainment in future. Therefore, we decide to build RoboCup System with standaolone robots under local communication constraint.

To reduce the computational load for the standalone robot, we propose to utilize color processing to identify the objects listed above. In other words, these object must be painted with different colors with which RoboCup player can easily identify the objects. Fig.6 and Fig.7 show examples of the painting of a field and a player.

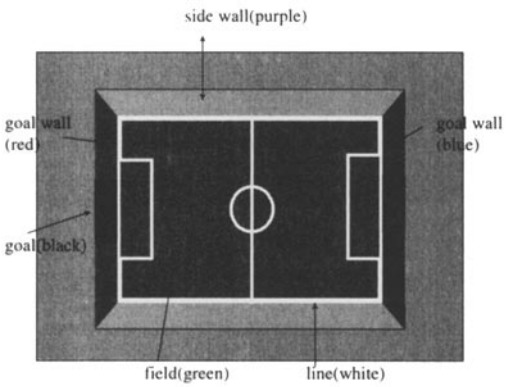

Fig. 6. Color Painted Field

We still need much computational power for the proposed color-painted RoboCup System to process the color image sequences and other tasks. There-

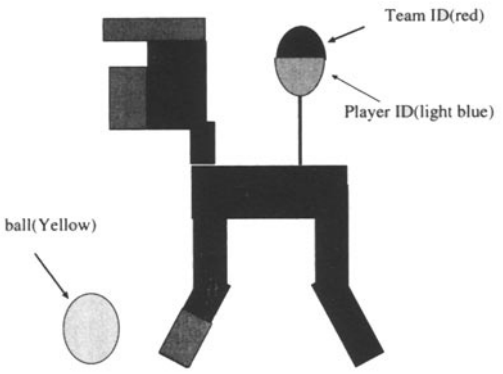

Fig. 7. Color Painted Player and Ball

fore, we developed a dedicated image processing engine to provide the required computational power. The details are described below.

It should be noted here that this field does not satisfy the RoboCup regulation defined by [4]. Furthermore, the size of the proposed robot described below does not satisfy the regulation, either.

4 Physical Robot Platform

4.1 Mechanical Configuration and Sensors

Fig.8 shows a mechanical configuration of the proposed platform. The features are as follows:

Robot Category: A quadruped legged walking robot.

Size and Weight: The size of the robot is about $110mm$ (length) $\times 90mm$ (width) $\times 110mm$ (height). The weight of the entire robot is about $1.3kg$.

Degree of Freedom (DoF): Three DoF for each leg and three DoF for neck and head.

Actuators: Each joint has a DC Geared Servo Motor which is composed of a DC coreless motor, gears, a motor driver, a potentio-meter, and a feedback controller. The torque of the geared motor is 4.2kgcm for the leg joints and 1.2kgcm for the neck-head joints. Each leg is designed to meet OPENR Configurable Physical Component (CPC) standard.

Image Sensor: A micro color CCD camera consisting of a lens module, a driver IC, a timing generator, A/D converter, and micro-controller is mounted on the head. The output signal format is 4:2:2 digital YUV. The image signal is downsampled into 360 x 240 pixel resolusion. The control signal is available through OPENR serial bus.

Microphone and Loudspeaker: A stereo microphone is mounted on the head, and the stereo signal is converted into the stereo digital 16bit with 32kHz sampling frequency. A monaural loudspeaker is mounted on the head, and 16bit DA converter through a speaker amplifier drives the loudspeaker. The audio module is connected to the OPENR serial bus.

Posture Sensor: Three acceleration sensors are mounted in the body to estimate the gravitation direction. These signals are AD-converted into 10 bit data with about 120Hz sampling frequency. This sensor module is connected to the OPENR serial bus.

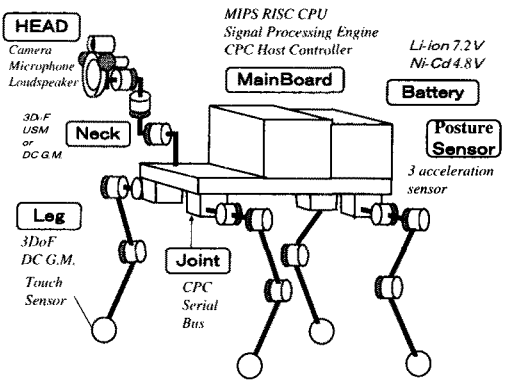

Fig. 8. Mechanical Configuration and Sensors

4.2 Hardware Architecture

The electronic hardware architecture is shown in Fig.9. The features are listed as follows:

CPU: A 64bit MIPS RISC CPU with 100MHz pipeline clock. The performance of the CPU is 133 MIPS.

Memory: 8MBytes SDRAM with 66MHz clock and 1MBytes Flash Memory. As Media for Program Storage in OPENR, we employ PCMCIA memory card.

ASIC-1:

Peripherals: Peripherals such as SDRAM controller, timer, and DMAC are integrated in a dedicated LSI for this platform. We pay attention to the bandwidth of the system bus because huge amounts of data, including the image signal is read from and written to the SDRAM. We design the DMAC in order for the CPU to be free from just copying the data.

Signal Processing Engines: This LSI includes a three layer multi-resolution filter bank for the CCD camera image, a color detection engine with eight color properties in YUV coordinate, an inner product engine for fast image processing, and a 16bit integer DSP with 50MHz for audio signal processing.

CPC Host Controller: We also include a host controller for OPENR CPC serial bus. All sensors are physically connected to this serial bus. (The image signal from the CCD camera is an exception.)

ASIC-2: Each component of the above CPC includes another dedicated LSI, the OPENR serial bus functional device. This LSI includes a single channel feedback controller for the DC servo motor, an interface for 16bit AD-DA converter, and four channel 10bit AD converter for general sensors. Each LSI has an EPROM for CPC information storage. This LSI works as a three port HUB for branching the serial signal stream.

Battery: A 7.2V Li-ion battery is utilized for the electrical circuits. Almost all electrical devices are operated with 3.3V, which is supplied via a DC-DC converter. A 4.8V NiCa battery is utilized for motor power. Since the current required for the motors is higher than the current limitation of the Li-ion battery, we separated the power supply into two batteries. Using these battery the physical robot works for about 30 minutes, which is an acceptable length for the RoboCup game.

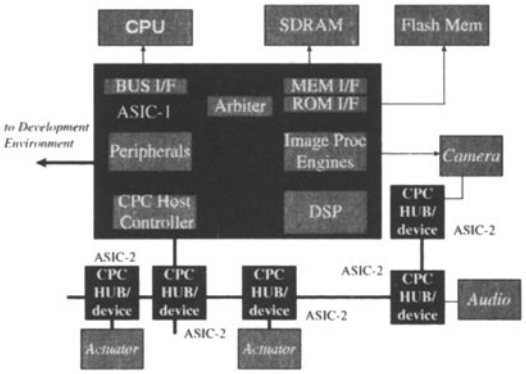

Fig. 9. Electronic Block Diagram

4.3 Software Architecture

We employ an object-oriented OS, Aperios[6], for the following reasons:

Object-Oriented Programming: Aperios supports the OOP paradigm from the system level. For communications among objects, several types of message passings such as asynchronous, synchronous, and periodic message passings.

Dynamic Loading: Originally Aperios supports an object migration protocol to accommodate heterogeneity. It is possible for the OPENR to use this migration mechanism in order to download application objects from the Development Environment to the robot Basic System. The downloading can be executed when the main application program is running.

In addition, we build some objects, such as the Virtual Robot and the Designed Robot defined in OPENR (Fig.10). The Virtual Robot supports APIs for communication with each CPC device, and the Designed Robot supports APIs for communication with a part of the robot with semantics such as left-front leg and head.

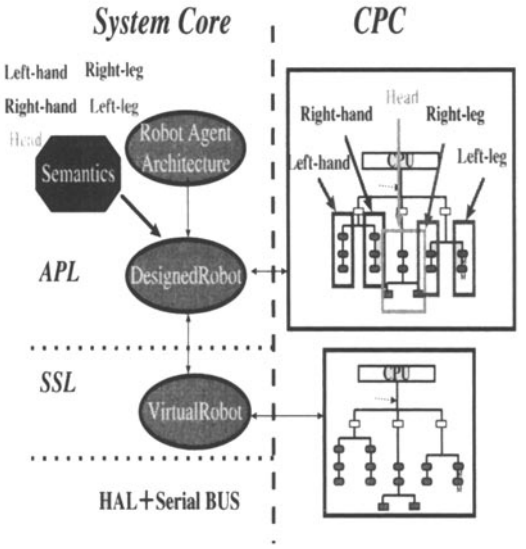

Fig. 10. Virtual Robot and Designed Robot

5 Algorithm Examples

In this section, we describe some examples of algorithms for (1) ball finding and kicking and (2) player positioning. These two algorithms are just examples, and

it depends on the strategy whether the above information is explicitly calculated or not. For example, kicking the ball may be considered as the hand-eye coordination problem[5]. Most of the algorithms have the explicit representation of the object position, however, if the sensor data has sufficient information to estimate the ball position, we can solve this problem without explicit representation of the ball position.

5.1 Ball Finding and Kicking

Since we define the colors of object exclusively, yellow color is assigned only for the ball (a yellow tennis ball is assumed to be used). In this case, to find the ball we just need to find a yellow object. To keep the ball in the center of the image, the robot head direction is feedback controlled and the walking direction is also controlled using the horizontal head direction.

Both the horizontal and vertical neck angle can be utilized to estimate the distance between the ball and the robot body (Fig.11). This information is also used as a queue to execute a kick motion.

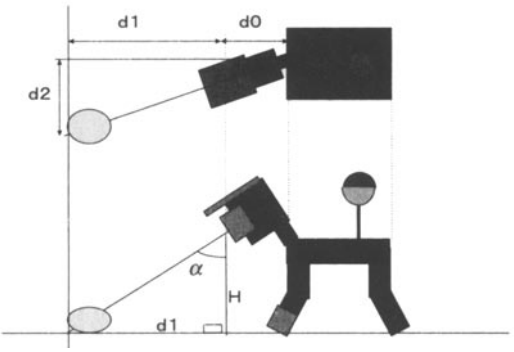

Fig. 11. Ball Positioning

5.2 Player Positioning

Here, the position means the two dimensional absolute position in the field, and we assume that the robot knows the absolute size of the field. It may not be necessary to measure the absolute position, however, it is easy to modify the algorithm into the relative position with relative information, namely, the proportion ratio of the width, length and height of the field, the wall, or the robot itself.

Assume that the robot knows H, the height of its camera view point from the ground. Then, it is almost the same problem as the estimation the distance between the ball and the robot.

1. To find the opponent goal side with the goal color,
2. to measure the neck's vertical angle along the edge of the downside of the wall,
3. to find the point that gives the minimum angle value x,
4. $Htan^{-1}(x)$ is the distance from the wall.

Repeating the same estimation of the distance to the side wall, the robot can get the two-dimensional position in the field.

There are many algorithms to estimate the robot position in the field, and it depends on the strategy of the game whether the accuracy of this information is important or not.

6 Conclusion

We proposed and developed a physical robot platform for RoboCup based on OPENR. The proposed RoboCup System, which is composed of the color painted field, the ball, and the robot players, does not satisfy the regulation defined by RoboCup Committee[4]. However, by assuming the color painted world, we can use the computational power for other tasks such as corporation and team play strategy.

In this paper we didn't describe the communication among robot players. We think that it is possible for this robot platform to communicate each other using the loudspeaker and the stereo mic. At least, signals from referee, such as a whistle, can be utilized as queues for the start of the game, and the end of the game, and so on.

In this paper we didn't describe the software development environment. This issue is very important for Robot Entertainment System, and as of the time of writing this paper, it is still under development.

OPENR will be able to supply more useful mechanical and electrical hardware and software components so that many researchers can share and test their ideas, not only for RoboCup but also for Robot Entertainment and for an Intelligent Robot Society.

7 Acknowledgement

The implementations of the proposed RoboCup System and robot platform are being performed by the members of Group 1, D21 laboratory. The MCM camera is developed by Sony Semiconductor Company, and Aperios is developed by Architecture Laboratory, Sony Corporation.

Authors would like to thank to Dr. Tani at Sony Computer Science Laboratory for his advice to the hand-eye coordination problem.

References

1. Achim, S. and Stone, P. and Veloso, M.: uilding a Dedicated Robotic Soccer System. Proceedings of IROS-96 Workshop on RoboCup. (1996) 41–48
2. Fujita, M. and Kageyama, K.: An Open Architecture for Robot Entertainment. Proceeding of the 1st International Conference on Autonomous Agents. (1997) 435–440
3. Kitano, H. : RoboCup: The Robot World Cup Initiative. Proceedings of International Joint Conference on Artificial Intelligence: Workshop on Entertainment and AI/ALife. (1995)
4. Kitano, H. and Asada, M. and Kuniyoshi, Y. and Noda, I. and Osawa, E.: RoboCup: The Robot World Cup Initiative. Proceedings of the First International Conference on Autonomous Agents. (1997) 340–347
5. Smagt, P. van der and Groen, F.C.A. and Krose, B.J.A.: A monocular robot arm can be neurally positioned. Proceedings of the 1995 International Conference of Intelligent Autonomous Systems. (1995) 123–130
6. Yokote, Y.: The Apertos Reflective Operating System: The Concept and Its Implementation. Proceeding of the 1992 International Conference of Object-Oriented Programing, System, Languages, and Applications. (1992)