Design of Real-Time Large Scale Robot Software Platform and its Implementation in the Remote-Brained Robot Project

S.Kagami Y.Tamiya M.Inaba H.Inoue Dept. of Mechano-Informatics, Univ. of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo, 113 Japan

Abstract

Constructing an environment for robot software research, which can program and experiment in various robot behaviors, design of a software development platform becomes important problem. This paper describes a design of "Software Platform" for real-time large scale robot software, and its implementation in the "Remote-Brained Robot Project". The Software Platform is designed as three layers, "MOTHER, BRAIN, SENSOR-MOTOR". MOTHER consists of tools to produce and evolve BRAIN programs.

For tools and libraries in such platform, there are two major problems. One is, there is tradeoff relationship between "Extension" and "Share". To overcome this problem, one course line was denoted. The other is, the methodology is needed between low-level realtime parallel environment monitoring program and

high-level software that takes much time.

A system developed for remote-brained robots and its application are described. This system aims at developing a many sorts of brain architecture and high level software that consists of various flexible multiprocess network.

1 Introduction

"Remote-Brained Approach [1]" is a paradigm of robot software research. In this approach, a robot brain is left in the remotely placed computer, and a robot body is connected to the brain through wireless link. In this paradigm, we could study intelligent robot architecture through building (1) a large scale and complex software such as learning, teaching, modeling, simulating, and (2) their parallel software environment.

Constructing such a system empirically from experiment, many developers should improve its software for the long term. Thus improvement process of the software system becomes important for realizing a real-world robot system.

In this paper, we propose "Software Platform" that could be shared by human developers, and the system itself evolves according to the top-level software. The

ware Platform".

There are big robot systems that control robot body from a large scale software such as vision processing and world model matching (or such kind of real-

top-level software will be an application of this "Soft-

world recognition program). (At the Hand-Eye system, COSMOS [2], Handy [3], and at the Mobile-Robot system, Navlab [4], Chatila [5], Yamabiko [6], Polly [7], etc.) However, these systems are an environment that is designed to treat specified simple body and its own task.

Recently according to the "Remote-Brained Approach", we developed many types of robot body (such as animal type and humanoid type) that has vision and another sensor, and improved both its top-level software [8–12] and its environment repeatedly. In this paper, we arrange this problem, and denote (1) Design of software platform for intelligent robot, (2) System composition problem of "share" and "extension", (3) Parallel environment monitor process system, (4) Event-driven interface between top-level robot program and low-level sensor process. Furthermore, an implementation for the remote-brained robot system and an experiment of adaptive behavior using humanoid type robot are described.

2 Design of Software Platform for Extensible Research Environment

2.1 Requirements of Software Platform

We define "Software Platform", that is a system to support robot brain program and provides (1) real-time programming facility of parallel sensor process and actuator control, (2) connection between real-time layer and non real-time layer, such as interface libraries of sensor and actuator control, robot body model libraries, robot action libraries and so on, (3) tools for developing a robot brain program (such as 3D simulator, brain architecture software, etc.), and (4) facility to customize itself according to evolution of top-level software. Then, structure of "Software Platform" is needed.

2.2 Structure of Software Platform Environment

Tools or environment for developing a robot brain is included in Software Platform. Therefore, we propose a structure that has three layers, "MOTHER" (tools or environment for developing a robot brain program), "BRAIN" (top-level robot program), and "SENSOR-MOTOR" (sensor processing and actuator control). (Fig.1)

Proc. IROS 96

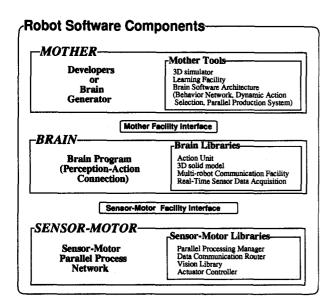


Figure 1: Robot Software Components

MOTHER is the environment to produce and evolve a BRAIN program (that generates robot behavior). This layer consists of two parts, (1) one is tools for developer to make robot motion or behavior, such as GUI or teaching tools, and (2) software to generate a robot behavior, such as 3D simulator, learning software, and brain software architecture library (SSA [13], BeNet [14], Dynamic Action Selection [15], Parallel Production System, State Transition Network, etc.).

The top-level software for robot behavior is placed at *BRAIN*. *BRAIN* contains robot behavior library, 3D solid model library of robot body or environment, multi-robot communication facility and so on.

SENSOR-MOTOR contains parallel sensor process and actuator control facility. Since this part generally consists of numbers of processors, parallel processing facility, task distribution handler, and inter-processor data communication are needed.

2.3 Tradeoff Problem of "Share" and "Extension"

When constructing a large scale software, "Share" and "Extension" become an important problem, however ordinary they have tradeoff relationship. Generally, object-oriented programing is used to enhance level of the "Share". Often object classes of fundamental facility are created, and gather them to construct a general class that is inherited many basic objects together and that has every "Method" of its parents and every "Slot Variable".

Nevertheless, changing a method function or slot variable exerts a strong influence not only to the object itself but also to all the class hierarchy system (all of its descendants). Thus creating a generic class object that inherits many fundamental class objects, is the possibility of causes to increase maintenance and management time. Furthermore, at group development, changing a basic object class is make stop to the top-level software development, then parallel development becomes impossible.

Then to balance "Share" with "Extension", class hierarchy of SENSOR-MOTOR layer should be flat and should not construct a generic class since this layer consists of numbers of processors and each library facility should work in parallel. On the other hand, class hierarchy of BRAIN layer is ordinary way.

2.4 Parallel Peripheral Monitoring Process

For BRAIN level (top-level) software, these three things are important.

- Facilities of sensor and actuator must be controlled transparent and flexible,
- Overhead of its system must be permissible and getting the sensor process results must be in realtime.
- Correspond to the top-level parallel program.

To realize a robot that behaves in real-world, two types of sensing facility are needed for the SENSOR-MOTOR part.

- 1. Sense designated sensor and return result,
- Monitor peripheral (For example: tracking a target on vision process) and return results continuously.

(1) is the sensing process that senses after top-level process has designated, and returns its result. (2) is the sensing process that senses regularly which had designated beforehand, and returns the results when the top-level program requires. The latter helps to raise the system reactivity. Since SENSOR-MOTOR is constructed from a multi-processor network, parallel sensor library, task distribution handler, interprocessor data communication facility and real-time sensor result trasmit facility are needed.

2.5 Interface Between BRAIN and SENSOR-MOTOR

There are two methods to return results of environment monitor process to the BRAIN (top-level) program.

- 1. Top-level program "asks" to SENSOR-MOTOR,
- Sensor process sets the results to the top-level program ("Event-Driven").

"Ask" method has two defects, one is that it is hard to process synchronously among sensor process and top-level program. Another is that communication overhead occurs when plural top-level programs refer to the results of one sensor process.

"Event-Driven" method enables many types of synchronous processing. Fig.2 shows the above.

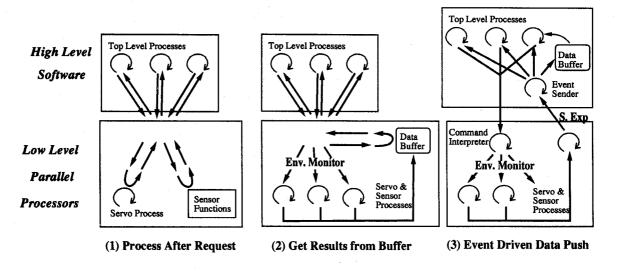


Figure 2: The 3 Methods to Get Sensor Process Results

3 Implementation of Software Platform for Remote-Brained Robot

3.1 Remote-Brained Approach

In "Remote-Brained" approach, a robot doesn't bring its own brain. A robot body has only actuator, sensor, battery and wireless transmitter/receiver. In this approach, brain and body of a robot are separated physically and connected through wireless link. There exist three layers, Brain, Body, and Interface between Brain and Body. With this approach, many advantages can be obtained to the research of intelligent robot software. That is;

- Brain and sensor processors have no physical limitations.
- It is possible to make an independent body that has many actuators (ex. humanoid type and many types of legged robot),
- Each layer can be developped respectively on the common interface,
- It is possible to exchange each layer,
- It is possible to share the brain and interface software.

This paper describes the software platform for Remote-Brained Robot system. Next section, each layer of software platform is described.

3.2 Brain Development Tool in MOTHER

MOTHER layer is a software to produce and evolve a BRAIN layer program (to appear [16]).

As tools for human developer, we are developing (1) GUI that calculates robot posture and body intersection, centroid and calculate robot view using 3D modeler, (2) behavior editor to edit motion sequence, and (3) automatic balancer to make humanoid type robot stable.

As tools for software, we are developing (1) brain architecture library such as BeNet [14], Dynamic Action Selection [15] and State Transition Network, and (2) 3D dynamic simulator to learn body motion.

3.3 3D Solid Model Body Library in BRAIN

Every robot body of remote-brained approach has its 3D solid model on BRAIN layer as a library. Since we adopted hobby radio control model parts as an actuator, and attached U-shaped parts to it, actuators were standardized and we made a 3D model library. We can make a body by connecting these parts, and 3D solid model of the body can be made of the same kind. Using this library, top-level software can calculate its intersection, centroid, path planning, and so

3.4 Parallel Sensing Controller for Environment Monitor Process in SENSOR-MOTOR

Since SENSOR-MOTOR layer is composed of plural processors, Parallel Sensing Controller (PSC) is needed to realize Environment Monitor. The PSC distributes given process to appropriate processors, and gather the results and send it to BRAIN Layer. Servo&Sensor Processes of Fig.2(2)(3) are the sensor processes that are started in parallel.

We developed a data communication router for transputer network. This router consists of 6 parallel processes and FIFO buffer. Each processor can be communicated synchronously in spite of physical position on the network. This router doesn't hang-up unless FIFO buffer is overflow.

3.5 Event-Driven Interface

between BRAIN and SENSOR-MOTOR

To get sensor process result using event-driven mechanism, the cycle of sensor process and top-level

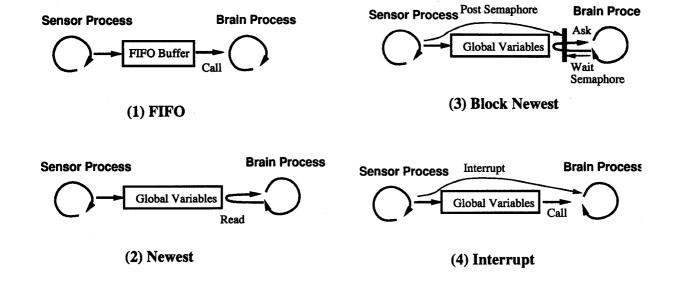


Figure 3: Event-Driven Data Returning to Brain Program

program becomes a problem. We implemented four methods (Fig.3),

- 1. Every results are transmitted (FIFO),
- 2. The Newest data are transmitted (Newest),
- 3. If the Newest data which have never transmitted exist, they are transmitted when top-level program requests, otherwise block top-level program (Block Newest),
- 4. Interrupt top-level program when the result data come in (Interrupt).

3.5.1 FIFO method

The FIFO buffer method transmits all of results from sensor process to top-level program. It is only available when a cycle of top-level program is equal or faster than sensor process. Otherwise (sensor process is faster than top-level program), queue becomes longer and FIFO buffer will overflow.

3.5.2 Newest method

This process overwrites the sensor process results, and transmits it by the request of top-level program. Thus, this method is available when a cycle of top-level program is equal or slower than sensor process. Otherwise (sensor process becomes slower than top-level program), processing the same data will occur. The implementation of the BeNet [14] adopts this method.

3.5.3 Block Newest method

Same as "Newest" method but to overcome the defect of the same data processing, only the data that have not passed to the top-level program are transmitted to the top-level program. Otherwise, top-level program is blocked until the fresh data arrive.

3.5.4 Interrupt method

When the top-level program cycle is relatively slow (such as model calculation or learning, etc.) and to react real-world input, sensor process results cause interruption in top-level program.

3.6 System Features

To implement for Remote-Brained robot system, we designed

- 1. Top-level interface is implemented in Euslisp/MT [17], (Object oriented parallel lisp including 3D solid modeler)
- 2. SENSOR-MOTOR layer is constructed from plural transputer based vision boards (about 20 boards) that have special chip of correlation, and data communication router is implemented,
- 3. Sensor process result can be returned to the toplevel software using Event-Driven method,
- 4. Vision facility can be used transparently by an object's methods and slots, and their substance is distributed for many processors. Their class hierarchy is wide toward the end to get "extension".
- 5. Multi-robot facility is implemented over an TCP/IP network,

Fig.4 shows an overview of dataflow of our system. Upper side is *BRAIN* layer and it shows components of the experiment of humanoid type robot's kicking behavior described the next section. (a) A Top-level program calls *SENSOR-MOTOR* libraries (to be called library object method), (b) command is transmitted to *SENSOR-MOTOR* layer by the form of S expression, (c) PSC decode the form and divide given task to the processors by calling correspond function, (d) sensor process results are seeded back periodically to

the top of SENSOR-MOTOR, (e,f) returning data is seeded to Euslisp/MT by the form of S expression.

4 Adaptive Behavior Experiment by Humanoid Type Robots

4.1 Humanoid Type Robot

According to the Remote-Brained Approach, we developed humanoid type robot body (called "Akira"). This robot has 16 D.O.F. (4 in each leg, 3 in each hand and 2 for neck), and it has TV camera in the head. Video signal is transmitted using Broadcast Satellite Channel (1.2GHz) to the SENSOR-MOTOR layer.

To develop a walking behavior, we implemented the Automatic Balancer using body model library (upper side of BRAIN layer in Fig.4). It checks the state transition. The states have only two modes, (1) both legs are attached to the ground, or (2) only one leg is attached. While the state mode doesn't change, it changes body's centroid to a designated point by "Stabilizer". Detecting the state to change, it generates motion from the old posture to the new posture by "Basic Motion Generator", and makes the robot stable.

4.2 Task: Ball Kicking

We tried to realize a ball kicking behavior (Fig.5). Initially, robot is placed about 500mm from the ball, and after starting, it follows the ball and walks in changing the direction until it reaches to the target. After that, it starts kicking motion.

Since the direction of the head changes a lot in walking motion, we control to keep looking for the target by calculating a direction and balance in real-time by "Posture Changer". In Fig.4, double-framed boy shows libraries of RR4 IN layer

time by "Posture Changer". In Fig.4, double-framed box shows libraries of BRAIN layer.

"Motion Selector" in Fig.4 gets tracking results by FIFO method and selects motion out of walking and turning. Reaching the ball, Motion Executer is stopped by Interrupt method and kicking motion is selected. "Posture Changer" gets data by Newest method.

5 Summary and Conclusions

In this paper, we proposed the "Software Platform" for real-time large scale robot software. The system could be shared by human developers, and the system itself evolves according to the top-level software. We denote four problems of such system, (1) Design of software platform for real-time large scale robot software, (2) System composition problem of "share" and "extension", (3) Parallel environment monitor process system, and (4) Event-driven interface between top-level robot program and low-level sensor process.

Using this platform, we developed many sorts of robot body (such as animal type and humanoid type) according to the "Remote-Brained Approach". Then we improved both top-level software and its environment repeatedly. Total system consists of over 20,000 steps in Lisp (at MOTHER and BRAIN layers), and over 8,000 steps in C (at SENSOR-MOTOR layer) except top-level robot behavior programs. Furthermore, an experiment of kicking behavior of humanoid type robot are described.

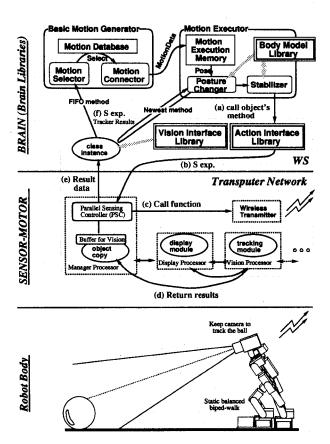


Figure 4: Dataflow between BRAIN and SENSOR-MOTOR during ball tracking behavior

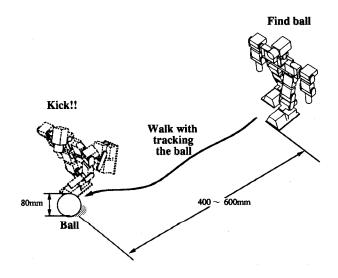


Figure 5: Ball Kicking Task

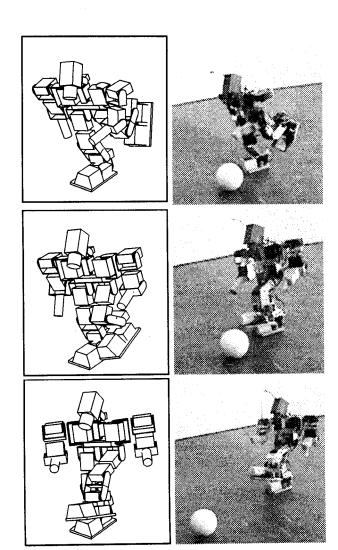


Figure 6: Ball Kicking Experiment by Humanoid Type Robot

References

- [1] Masayuki Inaba. Remote-Brained Robotics: Interfacing AI with Real World Behaviors. In Proceedings of the 6th International Symposium on Robotics Research (ISRR6); Robotics Research. The MIT Press, 1993.
- [2] T. Ogasawara and H. Inoue. COSMOS: A total programming system for integrated intelligent robot. Journal of the Robotics Society of Japan, Vol. 2, No. 6, pp. 507-525, 1984.
- [3] T. Lozano-Perez, et al. Handey: A Task-Level Robot System. In Hideo Hanafusa and Hirochika Inoue, editors, Proceedings of 4th International Symposium on Robotics Research, pp. 29-36. Published as Robotics Research 4, 1988.
- [4] Charles E. Thorpe, editor. Vision and Navigation :

- The Carnegie Mellon Navlab. Kluwer Academic Publishers, 1990.
- [5] Raja Chatila. Representation + reason + reaction → robot intelligence. In Proceedings of 6th International Symposium on Robotics Research (ISRR6), pp. 387– 397. MIT Press, 1993.
- [6] S. Suzuki, J. Iijima, and S. Yuta. Design and Implementation of an Architecture of Autonomous Mobile Robots for Experimental Researches. In *Proceedings of '93 ICAR*, pp. 423–428, 1993.
- L. D. Horswill. Polly: A vision-based artificial agent.
 In Proceedings of AAAI 1993, pp. 824-829, 1993.
- [8] Masayuki Inaba, Satoshi Kagami, Fumio Kanehiro, Koji Takeda, and Hirochika INOUE. Vision-Based Adaptive and Interactive Behaviors in Mechanical Animals using the Remote-Brained Approach. In Submitted to IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '94, 1994.
- [9] Masayuki Inaba, Fumio Kanehiro, Satoshi Kagami, and Hirochika Inoue. Two-armed bipedal robot that can walk, roll-over and stand up. In Proc. of Int. Conf. on Intelligent Robots and Systems, Vol. 3, pp. 297-302, 1995.
- [10] Masayuki Inaba, Satoshi Kagami, and Hirochika Inoue. Real-time vision plus remote-brained design opens new world for experimental robotics. In Preprints of the Fourth Int. Symposium on Experimental Robotics, pp. 68-73, 1995.
- [11] Masayuki Inaba, Fumio Kanehiro, Satoshi Kagami, and Hirochika Inoue. Vision-equipped apelike robot based on the remote-brained approach. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 2193-2198, 1995.
- [12] Masayuki Inaba. Extended vision with robot sensor suit. In Preprints of the International Symposium on Robotics Research, pp. 439-447, 1995.
- [13] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp. 14-23, 1986.
- [14] Tetsushi Oka. Function Integration for Autonomous Robots by Building a Network of Behaviors. PhD thesis, Graduate School of Engineering, University of Tokyo, December 1995.
- [15] Pattie Maes, editor. Designing Autonomous Agents (Theory and Practice from Biology to Engineering and Back). MIT / Elsevier, 1990.
- [16] Masayuki Inaba, Satoshi Kagami, Fumio Kanehiro, Ken'ichirou Nagasaka, and Hirochika Inoue. Mother Operations to Evolve Embodied Robots Based on the Remote-Brained Approach. In Proceedings of the 6th International Workshop on the Synthesis and Simulation of Living Systems, Artificial Life V. The MIT Press, 1996.
- [17] Toshihiro Matsui and Satoshi Sekiguchi. Design and implementation of parallel euslisp using multithread. *Information Processing Society*, Vol. 36, No. 8, pp. 1885–1896, 1995.