Proceedings of the 2nd Asian Control Conference, July 22-25, 1997, Seoul

Robot Arm Control System by Emergent Behavior Based Architecture

Yasushi Nakauchi, Youichiro Nishida, Takashi Shono and Yasuchika Mori
Dept. of Mechanical Engineering, National Defense Academy
1-10-20, Hashirimizu, Yokosuka 239, JAPAN
(E-mail: nakauchi@cc.nda.ac.jp)

Abstract

In order to utilize mobile robots in service fields, the robots should equip the functions for coexisting with human. In this paper, we discuss these required functions, and then propose EBBA (Emergent Behavior Based Architecture) to realize them. We also developed the robot arm control system based on EBBA. By experiments, we have confirmed the efficiency of EBBA.

1. Introduction

In order to utilize mobile robots in service fields, such as in hospitals, in offices and in houses, the robots should equip the functions for coexisting with human. The surrounding environments of such robots are shared by both robots and human. Thus, there is a danger that robots and human may physically collide with each other. Therefore, the robots must not injure the human. Human may also influence the shared environments. For example, when a robot is moving, human may move chairs or may put carton boxes on the floor. It causes the unexpected obstacles for robots. In general, obstacles are classified into two categories. One is the static obstacles and the other is dynamic ones. The static obstacles are the ones that are difficult to move and occupy the space eternally, such as walls, desks and bookshelves. On the other hand, the dynamic obstacles are the ones that may be moved and temporally occupy the space, such as human, other robots and chairs. Therefore, the robots should be able to avoid both of static and dynamic obstacles. Furthermore, if the robots stop their functions without being able to achieve given tasks, they won't be of any use. So it is required for the robots to be able to keep functioning until they achieve given tasks. To do so, the robots should have several methods to achieve given tasks and retry several times until they achieve the object. The required functions for human coexistent robots mentioned above are summarized as table 1.

Table 1: Required functions for human coexistent robots.

a) A robot must not injure human.
b) A robot must adapt to unknown environments.
c) A robot must keep functioning.

To realize such a robot, in our laboratory, we are developing human like mobile robot named WM (Work Mate). WM is equipped with dual arms, mobile robot base and CCD stereo vision camera system as shown in figure 1. WM is also equipped with multiple sensors to detect environmental information. They are touch sensors, infrared sensors, ultrasonic sensors and CCD cameras. To utilize WM as a human coexistent robot, we need a suitable architecture to integrate these systems.

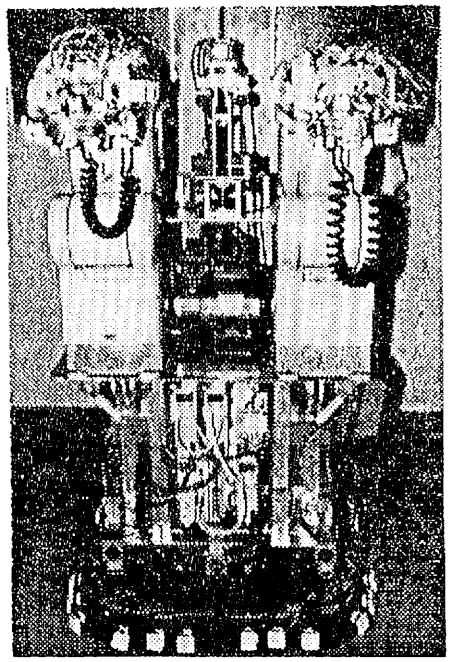

Figure 1: The out look of WM (Work Mate)

In this paper, a new architecture named EBBA (Emergent Behavior Based Architecture) is proposed for constructing intelligent robots. The paper is organized as follows. In section 2, architecture of EBBA

is described, and in section 3, the robot arm control system based on EBBA is described. Then in section 4, the experimental results are presented.

2. EBBA

EBBA is designed to realize required functions for human coexistent robots as shown in table 1. At first, to realize a robot that doesn't injure human, a system must be sensor based architecture by which obstacles and human are detected without failure. Seconderly, to adapt to unknown environments, the system should respond to the changes of the environment in realtime. Therefore, the system must be constructed by a set of concurrently executable behavior modules. Thirdly, in order the system to keep functioning, it should have several methods to achieve given tasks. Also these methods must be controllable by a planner, so that it can emerge behaviors along with the situation.

These required functions of EBBA is summarized as shown in table 2. We have designed EBBA based on these requirements.

Table 2: The basic ideas for designing EBBA.

i)	Sensor based architecture.
ii)	Constructed by a set of concurrently executable behavior modules.
iii)	To have multiple methods to achieve given tasks by utilizing behavior modules.
iv)	A planner can control emergent behaviors.

2.1. Behavior Module of EBBA

EBBA is constructed by behavior modules as shown in figure 2. Each behavior module is concurrently executable entity which has its own object state (OS) and internal states (ISs) as internal memory. An OS is the state that represent the current object of a behavior module. ISs are the parameters that effect the action of a behavior module. Each behavior module refers not only sensor inputs but also its own and other modules' OSs and ISs, then decides its actions. As the resulting actions, each behavior module can modify its own OS and ISs, can modify other behavior modules' OSs, and can output orders to actuators.

Even if the same information is detected from sensors, the behavior of each module can be different depending on its OS and ISs. These actions derived by the interaction with environments are viewed as emergent behaviors. In order to control emergent behaviors of a system, a planner modifies other behavior modules' OSs.

2.2. System Control Architecture of EBBA

A typical system control architecture based on EBBA is the one as shown in figure 3. Within the framework of EBBA, a planner is also realized as a behavior module. A planner behavior module is different from others in the way it inputs other module's OSs as its sensor inputs. Thus, a planner can monitor the system's emergent behavior[1]. If it is necessary, a planner can control the system's emergent behavior by just modifying other behavior modules' OSs.

Furthermore, with EBBA, we can construct meta-planners that control planners. A planner also has its own OS and ISs as shown in figure 2. Therefore, meta-planners are realized by just connecting their sensor inputs to other planners' OSs and ISs. This architecture is suitable for constructing more complicated and larger scaled system. For example, we can construct planners for each of robot hand, mobile robot base and eye system, then we can integrate them by using a meta-planner that controls these planners. In this way, we can expand a system very easily.

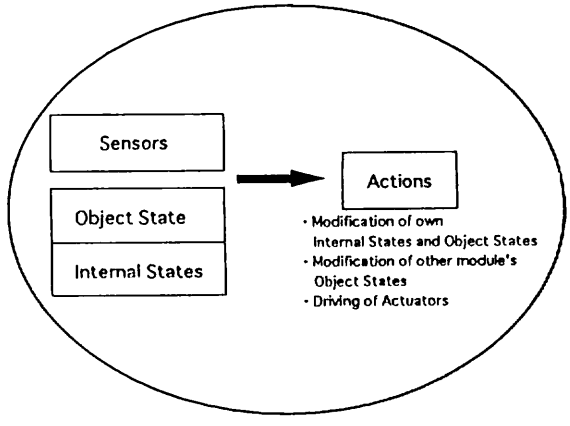

Figure 2: A behavior module of EBBA.

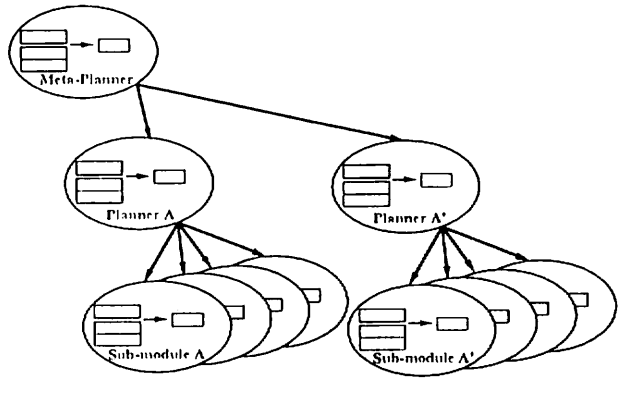

Figure 3: A typical system architecture based on EBBA.

A behavior module we have described in 2.1 is simi-

[1] In this paper, we define an emergent behavior as the behavior that is derived and realized by the combination of multiple behavior modules.

lar to the one in SSA[1], in the way it acts by referring sensor inputs. But it is different in EBBA as to that the actions of each behavior module differ by its OS. Thus, a planner can control emergent behavior of the system by modifying the lower level behavior modules' OSs.

The system architecture of EBBA is rather simple. A system designer prepares necessary behavior modules, then connect reference lines to other modules' OSs and ISs from one module to another. If there are multiple modification lines go into one module, the priorities[2] of the lines can be set.

3. System Configuration

3.1. Robot Arm Control System

We have designed robot arm control system based on EBBA as shown in figure 4. The OSs that each behavior module may take are shown in table 3. The arrows in figure 4 denote the read/write lines to other modules' OSs and ISs. Each behavior module is illustrated as a layer. In the lower level (level 0, 1, 2, 3), we arranged basic modules that must be realized to avoid system failure. On the other hand, in the heigher level (level 4, 5, 6), we arranged the modules to achieve given goals. These heigher level modules refer and control the lower level modules' OSs and ISs.

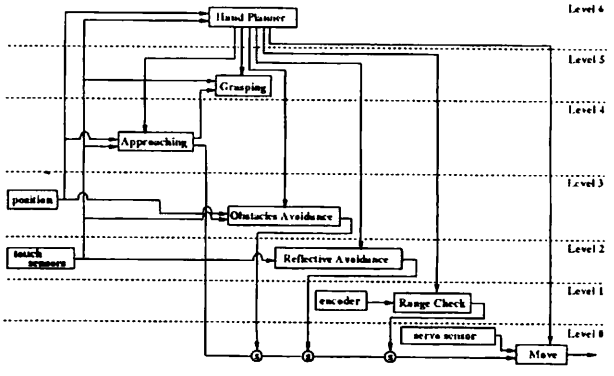

Figure 4: Control architecture of robot arm.

3.2. Emergent Behaviors

Assume that a human is extending his/her hand towards a cup for grasping it. If some obstacles touched on hand or arm before s/he reached the hand to the cup, s/he may detour the hand around the obstacle, s/he may twine the the hand around the obstacle, or s/he may push the obstacle aside. These three kinds

[2]The priorities are depicted as subsuming lines in figure 4.

Table 3: Object state of each behavior module.

Level	Behavior Modules	OS (Object State)
0	Move	linear mode
		joint mode
1	Range Check	movable range check
2	Reflective Avoidance	stop & backward
		stop & forward
		stop
3	Obstacles Avoidance	inclined direction
		forward
4	Approaching	towards object
		approach direction
5	Grasping	open
		close
6	Hand Planner	goal
		idle

of avoidance behaviors can be realized by setting level 0, 3 and 4's OSs as shown in table 4.

In this way, by just changing parameters (object state) of behavior modules, we can realize many kinds of emergent behaviors. Therefore, a planner can control these emergent behaviors by changing the lower behavior modules' OSs. But in the framework of SSA and the previous works, theses parameters are not explicitly expressed and utilized. Therefore, even if we want to add system's behaviors that might be expressed by the combination of behavior modules prepared so far, we have to add behavior modules for each behaviors. Thus, the number of behavior modules will be immense. And this cause the difficulty for a planner to manage behavior modules.

4. Implementation on WM

We have implemented the robot arm control,system on WM. The hardware structure of WM's robot hand system is as shown in figure 5.

We have used two Mitsubishi RV-E2s as the 6-axes dual arms. Around the surface of the robot arm and hand, we have covered them by using 92 touch sensors for each arm (see figure 1). As main CPU, we used VME CPU board AVME-130 on which the realtime OS VxWorks runs. And via AVME-350 serial I/F board and a parallel port of AVME-130, all the sensors and the servo motors of arm robots are connected. We also used SUN SS5 for developing and downloading programs.

We have implemented the system control architecture described in section 3 as follows. Each behavior module of EBBA is written as a function of C program. And each function is implemented as a concurrently executable task of VxWorks. Therefore, by using the realtime scheduling scheme of VxWorks, we

Table 4: Emergent behaviors derived by same behavior modules.

	Detouring Avoidance Emergent Behavior	Pushing Avoidance Emergent Behavior	Twining Avoidance Emergent Behavior
Move Behavior	It moves a hand by linear mode.	It moves a hand by linear mode.	It moves a hand by joint mode.
Reflective Avoidance Behavior	If something touchs on sensors, it stops and moves a hand backward a little.	If something touchs on sensors, it stops and moves a hand forward a little.	If something touchs on sensors, it stops a hand.
Obstacles Avoidance Behavior	It moves a hand $5cm$ towards the 90 degree inclined direction where something touched.	It moves a hand $5cm$ towards the direction where something where something touched.	It rotates a movable joint 5° towards the sub-goal position.

insured the sampling time of sensors.

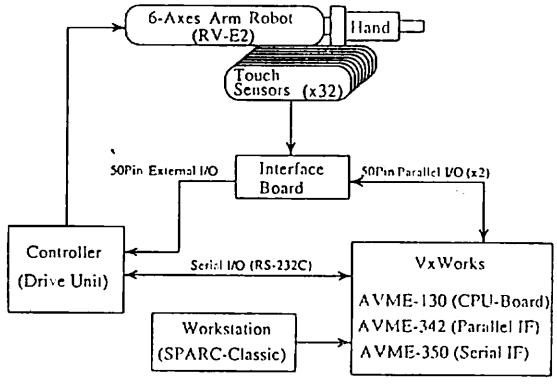

Figure 5: Hardware structure of the robot hand system.

system based on EBBA can avoid both static and dynamic obstacles.

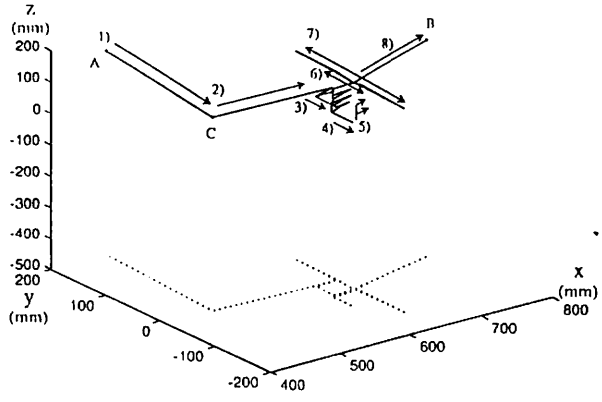

Figure 6: Experimental results.

5. Experimental Results

To confirm the efficiency of EBBA, we have evaluated the system in the various situations. One of the experimental results is shown in figure 6.

We assumed the planner knows the environmental map beforehand and is given goal positions (A, B and C in the figure). In the figure, the locus of the top hand is illustrated. At first, the hand planner set the systems emergent behavior as detouring avoidance, so the detouring avoidance emergent behavior is emerged when the arm touched an obstacle (between C and B). The planner watched it is taking too many retrials by referring the counter IS of level 2 behavior, then it decided to change the emergent behavior to the pushing avoidance. Thus the 5th time the arm touched the obstacle, it pushed the obstacle aside. And finally, it succeeded to reach the goal position B.

We also experimented by pushing arm by our hand. While we were pushing the arm by our hand, the arm went back (reflective avoidance behavior) until we stop it. And when we quit it, the arm resumed for extending hand to the goal position (approaching behavior). In this way, we have confirmed that the arm control

6. Conclusion

In this paper, we proposed EBBA the new architecture for constructing intelligent robots. And we have designed and implemented the robot arm control system based on EBBA. By experiments, we have confirmed that the system based on EBBA satisfies the required functions for human coexistent robots.

We are currently developing the mobile base navigation system[2] and the stereo vision system of WM by using EBBA. We are planning to integrate them as one service robot in near future.

References

[1] R. Brooks, "A Layered Intelligent Control System for a Mobile Robot", *IEEE J. Robotics and Automation*, Vol.RA-2, No.1, pp.14-23, 1986.

[2] Y. Nakauchi and Y. Mori, "Emergent Behavior Based Sensor Fusion for Robot Navigation System", *Proc. Itnl. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp.525–532, 1996.