濟州大學校 尖端技術硏究所 論文集, 第16卷, 1號, pp.53-59, 2005. 6
Cheju Nat'l Univ. Res. Inst. Adv. Tech. Jour., Vol.16. No.1, pp.53-59, 2005. 6

실버메이트 로봇을 위한 행위기반의 하이브리드 제어구조 연구

최 경 현* · 도 양 회**

A STUDY ON BEHAVIOR-BASED HYBRID CONTROL FOR ELDERLY ASSISTANT ROBOT

Kyung-Hyun Choi* · Yang-Hoi Doh**

ABSTRACT

Elderly assistant robot is becoming more important with the progress of medical technology and increasing aging society. An elderly assistant robot can help the elderly and the disabled who have difficulties doing daily life independently in an apartment and in silver town. It is a kind of mobile manipulator robot that can manage various housekeeping and home care. To accomplish these various and complex function, hybrid control has been suggested as a mutually complementing architecture of the weak points of a deliberative and a reactive control. This paper addresses hardware and software control architecture, and a methodology of task representation to accomplish various and complex tasks successfully. Hardware control architecture is decentralized. Software control architecture consists of three layers of deliberative, sequencing, and reactive as hybrid control architecture. Multi-layer behavior model is employed to represent desired tasks.

Key Words : intelligent robot, hybrid control, silvermate robot

Ⅰ. INTRODUCTION

With the development of medical technology, the average age is growing older. Accordingly, the importance of elderly assistant robot has been increased. Elderly assistant robot is a kind of mobile manipulation robots to help housekeeping, home care, and leisure activities for the elderly who have difficulties doing daily life independently in such environments as an apartment or silver town.

For optimal performance of the given tasks an systems elderly assistant robot requires intelligence control like human brain, for example, autonomous function based on the intelligence control system, man-machine-interface for communication between man and robot, sensor function using highly

* 제주대학교 기계에너지시스템공학부, 첨단기술연구소
Department of Mechanics Eng., Res. Inst. Adv. Tech., Cheju Nat'l Univ.
** 제주대학교 전기전자공학부, 첨단기술연구소
Department of Electrical & Electronic Eng., Res. Inst. Adv. Tech., Cheju Nat'l Univ.

developed sensors, movement function for extending working area, manipulation function to accomplish given tasks, and processing function for mission accomplishment. To execute the various and complex tasks, improvement not only in mechanical system architecture but also in control system architecture is needed.

Existing deliberative control system is a traditional hierarchical layering control architecture that plans and executes path with reasoning function based on intelligence-base robotics. Generally, it executes optimal performances in a complex task, however, it has weaknesses of its slow reactive speed and dependence on outer environment model. Behavior-based reactive control system defines reactions of actuator finitely with sensors and processes real-time robot performances. However, problems may occur from minor details and weaknesses can be found in general task performances. Reactive/deliberative hybrid architecture is needed as a mutually complementing architecture of these weaknesses [1-5]. The suggested hybrid control architecture consists of three layers such as deliberative, sequencing and reactive layers. The deliberative layer has the function of interfacing with a user and executing task planning. The sequencing layer is classified into two groups. The first group is the controlling part that executes the internal process by managing the components in the reactive layer. The second group is the information part that extracts highly advanced information from raw data using various kinds of sophisticated and time-consuming algorithms. The reactive layer controls robot motion by executing relatively simple computation in real-time.

This paper suggests distributed hardware control architecture, behavior-based control architecture, and a multi-layer behavior model representing a way of task expression.

II. HARDWARE CONTROL ARCHITECTURE

Mobile of an elderly assistant robot is the form of two active casters. A manipulator has 7 DOF arms and three fingers of each 3 DOF. A trailer can be equipped to increase its function. Fig. 1 shows hardware control architecture of the proposed elderly assistant robot. It shows distributed control architecture and consists of independent control systems of two PCs [6]. Linux is employed for OS for real-time processing. Each PC communicates each other based on Ethernet, and the distributed control modules are connected to PC by CANbus.

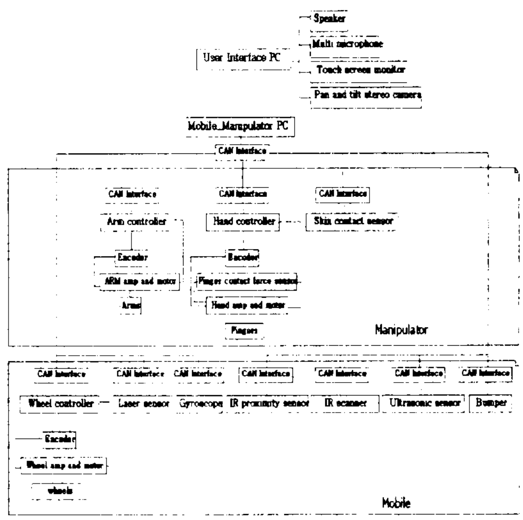

Fig. 1 Proposed hardware architecture

III. SOFTWARE CONTROL ARCHITECTURE

3.1. Behavior-based Hybrid Control

The concept of behavior appeared in subsumption architecture from Brooks in mid-80s as an alternative to the weak points of SPA(Sense-Plan-Act)[7]. SPA has its weaknesses in the sense that it requires a lot of information about task environment beforehand, and it cannot deal with uncertain and unpredictable environment effectively.

This control method divides the components according to their abilities contrary to the existing control method, which divides the components based on their functions. This concept has been expanded and behavior-based control is generated from it. Behavior-based control has reliability in autonomous responding to the environment of intelligent agent. However, it has weak points in a given task performance. This is because cautious task planning and management are excluded from behavior-based control.

To solve these problems a study on hybrid architecture that has deliberative layer and reactive layer concurrently began from early 90s. Generally, in the upper architecture of hybrid control architecture are the parts in charge of task decomposition and planning and deliberative layer related to HRI(Human Robot Interface). In the lower architecture of hybrid control are reactive layer for real-time robot-control and sensing. A layer in charge of sequencing can be placed between the two layers. Therefore, hybrid control architecture needs at least two layers, and usually it is formed with 3 layers.

Fig.2 shows the proposed hybrid architecture. The proposed architecture has three layers such as deliberative, sequencing, and reactive layers and Knowledge database.

The deliberative layer has the function of interfacing with a user and executing task planning, and it consists of HRI and task planner. The sequencing layer is classified into two groups. The first group is controlling part that executes the internal process by managing the components in the reactivelayer, and it consists of process supervisor and process storage. It generates and modifies process and sends behavior to reactive layer. Also, it reports the performance results of the reactive layer to HRI. The second group is information part that extracts highly advanced information from raw data using various kinds of sophisticated and time-consuming algorithms. This group is divided into the following four related parts: navigation modules, manipulation modules, and speech/object recognition modules. Reactive layer is composed of modules related to hardware, for example, sensor and actuator, and behavior generating real-time control command.

Knowledge database is formed with user knowledge, task knowledge, and world model. User knowledge has storage of user information, and task knowledge has required task storage expressed in a multi-layer behavior model. World model has storage of environment, maps and images. Knowledge database can be updated with modification and addition of data in off line through knowledge editor.

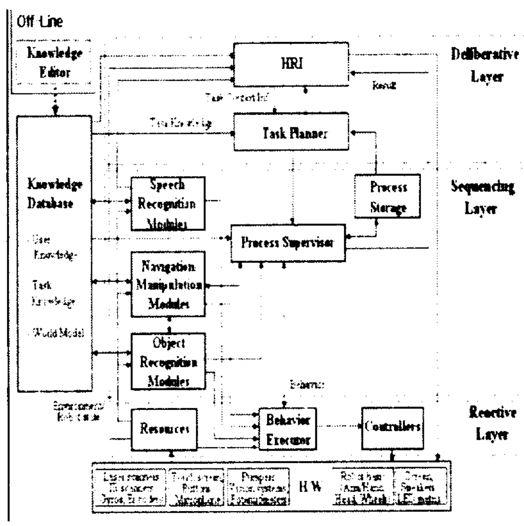

Fig. 2 Proposed hybrid control architecture

3.2. Deliberative Layer

Deliberative layer consists of HRI and task planner. Fig. 3 shows the deliberative layer.

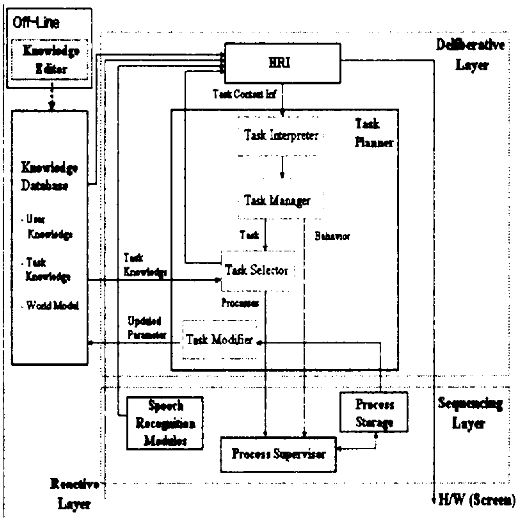

Fig. 3 Deliberative layer

HRI receives commands from the users through communication or touch screen and sends task commands to task planner. HRI also expresses internal condition of a robot.

Task planner is in charge of task planning and consists of task interpreter, task manager, task selector, and task modifier as its internal components. Task interpreter interprets the task context received from the HRI. task manager classifies the interpreted context information into simple task and complex task, and then it sends simple task to process executor of sequencing layer and complex task to task selector. Simple task denotes that a task does not require process, for example, a user call. complex task denotes that a task requires process such as cleaning a room.

Task selector selects required tasks for the complex task from task knowledge database and sends process related tasks to process generator. Task modifier updates the modified path and trajectory of process storage to the task knowledge database after the completion of process performance.

3.3. Sequencing Layer

Sequencing layer controls the reactive layer and generates information. Process supervisor generates and modifies process and sends behavior to reactive layer. Also, it reports the performance results to HRI. process generator and process executor form the process supervisor as internal components. Fig. 4 shows the sequencing layer. Process generator stores processes received from task planner in the process storage. Then the process generator generates process based on the world model knowledge database and sends it to the process executor. The process executor evaluates the internally generated behavior in the process and the behavior directly received from the task planner. Then, the process executor classifies them into the behavior requiring navigation and manipulator modules and the behavior that is to be performed in reactive layer, and then reports the results to HRI. Also, the process executor requests that the process generator modify the process according to events and guard conditions.

Process storage stores the processes. The modified path and trajectory are stored in task knowledge database from task modifier. The process storage is deleted after the completion of process.

Navigation modules update the world model knowledge database through map building, or performs localization and path planning referring to the world model knowledge database and the present robot condition. Generated path is sent to the process executor or the behavior executor of the reactive layer.

Manipulation modules generate point-to-point or continuous trajectory to the related behavior according to the world model knowledge database and the present robot condition.

Speech recognition modules process user voice data received from resources referring to the knowledge database and send the recognition results to HRI.

Object recognition modules process objects and user face recognition data received from resources referring to knowledge database and send the recognition results to navigation / manipulation modules, behavior generator, and process executor. Object recognition modules also update new data and modified data to the knowledge database.

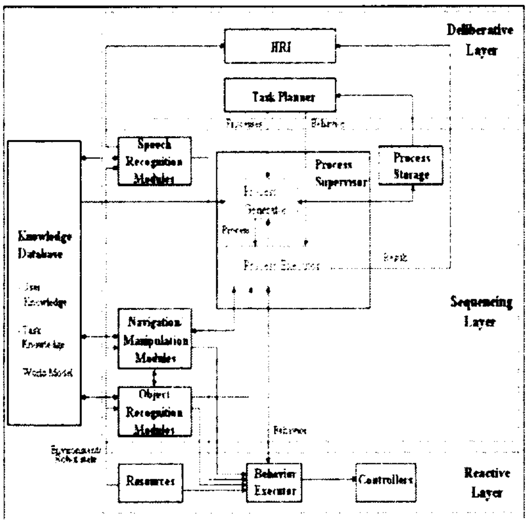

Fig. 4 Sequencing layer

3.4. Reactive Layer

Reactive layer consists of hardware related modules such as sensor, actuator, and behavior generating real-time control commands. Fig. 5 shows reactive layer. Resources collect all the input data from each sensor and make it easy to share the data. Behavior executor sends each behavior received from process executor to each controller, and it connects sensor and actuator. Behavior executor has basic behaviors and collision avoidance as its internal components. Controllers send direct commands to each actuator of a robot.

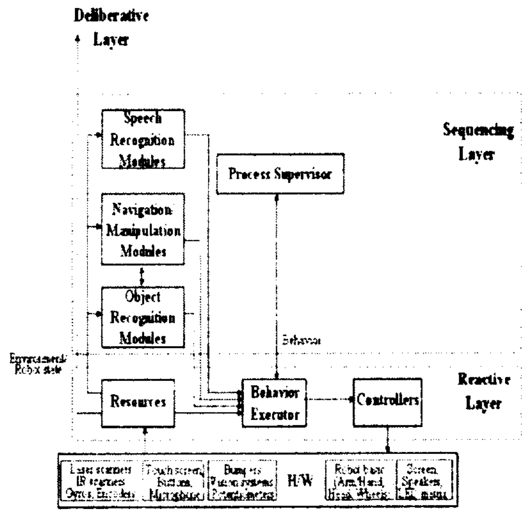

Fig. 5 Reactive layer

Ⅳ. TASK REPRESENTATION

Task is expressed in a multi-layer behavior model as shown Fig. 6. Task is shown as a collection of behavior and each behavior has sub-behaviors. Transition between behaviors is defined by events and guard conditions. Process is a sequencing order of behaviors that express tasks with the multi-layer behavior model, and the process includes error recovery logic. Task has simple process or complex process. The task processes expressed with the multi-layer behavior model are stored in the task knowledge database.

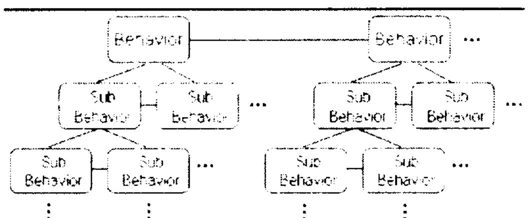

Fig. 6 Multi-layer behavior model

Fig. 7 is a statechart diagram using UML to Delivery, the basic task. Each state denotes behavior. Fig. 8, Fig.9, and Fig. 10 show the multi-layer that robot moves continually to a goal position after opening and closing doors if robot detects a closed door while performing Move command.

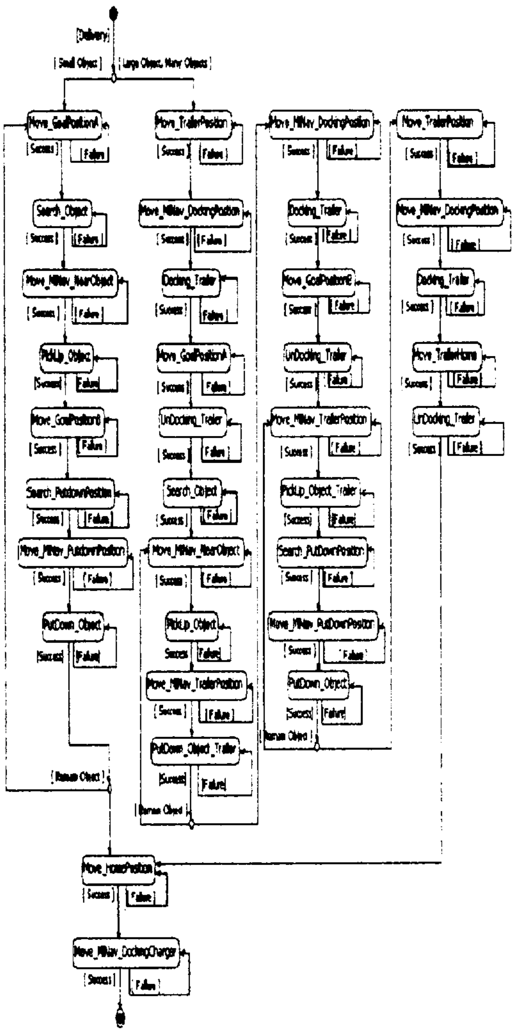

Fig. 7 Statechart diagram of delivery task using UML

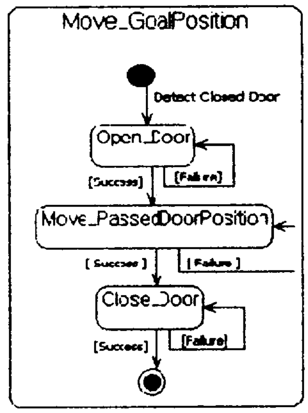

Fig. 8 Statechart diagram of sub-layer for Move command

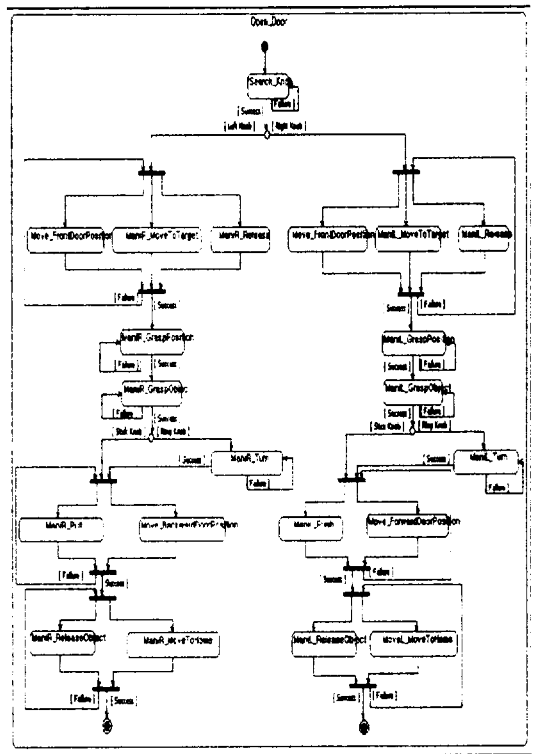

Fig. 9 Statechart diagram of sub-layer for Open_Door Command

Part A in Fig. 9 shows synchronized behaviors of navigation and manipulation to open a door. This enables the movement more natural and faster.

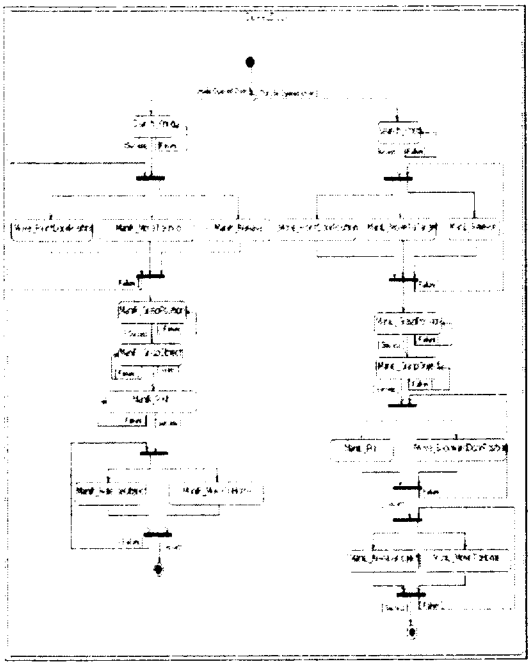

Fig. 10 Statechart diagram of sub-layer for Close_Door command

V. CONCLUSION

This paper proposed a distributed hardware control architecture of an elderly assistant robot and behavior based control architecture formed with deliberative, sequencing, and reactive layers. Elderly assistant robot is for the elder who have difficulties doing daily life independently in an apartment or in silver town. Also, a multi-layer behavior model is proposed as a way for task expression.

The distributed hardware control architecture can increase reliability of an elderly assistant robot, and the behavior-based hybrid control architecture can control software quickly and accurately. The multi-layer behavior model can define tasks effectively.

In the future, the proposed control architecture needs to be applied to real robot platform, and each task should be defined using UML with the multi-layer behavior model. Finally, realization of this elderly assistant robot is needed.

VI. REFERENCES

4) Reid G. Simmons, "Structured Control for Autonomous Robots," IEEE Transactions on Robotics and Automation, Vol. 10. No. 1, pp.34-43, 1994.
2) R. C. Arkin, "Behavior-Based Robotics," The MIT Press, Cambridge, Massachusetts, London, England, pp.205-235, 1998.
3) Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, and Dieter Fox, "MINERVA : A Second-Generation Museum Tour-Guide Robot," Proc. of the IEEE Conference on Robotics and Automation, Detroit, Michigan, USA, pp.1999-2005, 1999.
4) Mattias Lindstrom, Anders Oreback, and Henrik I. A Christensen, "BERRA : A Research Architecture for Service Robots," Proc. of the IEEE Conference on Robotics and Automation, San Francisco, CA, USA, pp.3278-3283, 2000.
5) Gunhee Kim, Woojin Chung, Munsang Kim, and Chongwon Lee, "Implementation of Multi-Functional Service Robots Using Tripodal Schematic Control Architecture," Proc. of the International Conference on Robotics and Automation, New Orleans, LA, USA, 2004.
6) A. Traub, R. D. Schraft, "An Object-Oriented Realtime Framework for Distributed Control Systems," Proc. of ICRA-99, pp.3115-3121, 1999.
7) Rodny A. Brooks, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, Vol. 2, No.1, pp. 14-23, March 1986.each other based on Ethernet, and the distributed control modules are connected to PC by CANbus.