5

Development of Intelligent Service Robotic System Based on Robot Technology Middleware

Songmin Jia, Yoshiro HADA, Takayuki Ohnishi,
Harunori Gakuhari and Kunikatsu Takase
University of Electro-Communications
1-5-1 Chofugaoka, Chofu-City, Tokyo 182-8585,
Japan

1. Introduction

Many service robotic systems have been developed in order to improve care cost and the QoL (Quality of Life) of the elderly people in the population-aging society (Nagi, N. Newman et al., 2002; Schulz, D. et al., 2002; Stein, M. R. Stein et al. 2000). Our HARSP (Human-Assistance Robotic System Project) project has been developing a network distributed Human-Assistance Robotic System since 2000 (Jia et al., 2001; Jia et al., 2004). We developed hardware base, key technologies, and implemented CORBA (Common Object Request Broker Architecture) application servers to provide some basic services to aid the aged or disabled. Improvements of the developed system have been doing from practical viewpoint. In order to enable remote users to get a better understanding of the state of robots working and the state of the aged or disabled, we developed network distributed monitoring system to transmit media streams in real time. Home robot integration system and network monitoring system using QuickCam Orbit cameras were developed and demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan. The omnidirectional wheelchair and maneuvering system have been developed to enable the disabled person to operate it skilfully. The developed powered wheelchair gives users the same degree of mobility that healthy individuals enjoy (Ohnishi et al., 2002). This new wheelchair can be operated by a disabled wheelchair user using either a joystick or by simple body actions with two hands free. Force sensors arranged under the seat detect the dynamic changes in the user's posture in the wheelchair. We also developed the omnidirectional intelligent service robot which can bring the necessary objects such as a canned drink, a towel or a newspaper to wheelchair user. In order for the service robot to perform a task autonomously, iGPS (indoor Global Position System, Hada et al., 2004) was developed to localize the mobile robot. In addition, a network distributed monitoring system was developed in order to observe the state of the robotic system, as well as the state of the user. In the developed system, Robot Technology Middleware framework was used to develop robotic functional elements as RT components that can facilitate network-distributed software and sharing. AIST (Agency of Industrial Science and Technology, Japan, http://www.is.aist.go.jp/rt/) developed RTM to promote application of Robot

Technology (RT) in various fields (Ando et al., 2004). Modularization of robot components is a key technology for the efficient construction of robot systems. RTM is based on CORBA (omniORB), so the components of the system can be implemented by different programming languages, run in different operating system, or connected in different networks to allow interoperation. We develop robotic functional elements as "RT component", which makes application and system integration easier. It is very easy for the user to create new application system by re-using existing RT components, thus lowers the cost of development of new robotic system. This paper introduces the architecture of the developed system and the results of experiments demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan.
The rest of the paper consists of 5 sections. Section 2 describes concepts of Robot Technology Middleware (RTM) developed by AIST. Section 3 presents the developed home integration robot system. Section 4 introduces the developed network monitoring system using QuickCam Orbit cameras. The experimental results are given in Section 5. Section 6 concludes the paper.

2. Robot technology middleware

Robot Technology Middleware (RTM, Kitagaki et al., 2004) was developed by Agency of Industrial Science and Technology of Japan (AIST) to promote application of Robot Technology (RT) in various fields. RTM aims to modularize robotic functional elements as "RT software component", which enables the system to be extended and integrated easier for a new system or new applications. In order to enable system to be language independence, operating system independence, RTM has been developed based on CORBA. CORBA (Common Object Request Broker Architecture, Object Management Group, http://www.omg.org) is a distributed computing technology. There are the other distributed middleware like RMI (Remote Method Invocation, (Java remote method invocation, http://java.sun.com/products/jdk/rmi/index.html), DCOM (Distributed Component Object Model), MOM (Messages Oriented Middleware, Message-orientated middleware: http://sims.berkeley.edu/courses/is206/f97/GroupB/mom/). In contrast to all of these, CORBA uses an Object Request Broker (ORB, Object Oriented Concepts, Inc., http://www.omg.org) as the middleware that establishes a client-server relationship between objects, and it is an object-oriented extension of Remote Procedure Calls (RPCs). CORBA uses GIOPs (General Inter-ORB Protocols), ESIOPs (Environment Specific Inter-ORB Protocols) and IIOP (Internet Inter-ORB Protocols) to implement a truly heterogeneous distributed system. This heterogeneity enables CORBA to inter-operate ORBs purchased from different vendors and supported on different platforms. RTM used OminORB 4.0.5 to implement framework. OmniORB is a robust, high-performance CORBA 2 ORB, developed by AT & T. It is one of only three ORBs to be awarded the Open Group's Open Brand for CORBA. This means that omniORB has been tested and certified CORBA 2.1 compliant. OmniORB implements the specification 2.3 of the Common Object Request Broker Architecture (CORBA).

2.1 Structure of RTM

RTM includes Standard RT Components, Standard RT Services, RT Components Framework and RT Library (RTM::RtcBase, RTM::InPortBase, RTM::OutPortBase and so on). Figure 1 shows the structure of RT middleware.

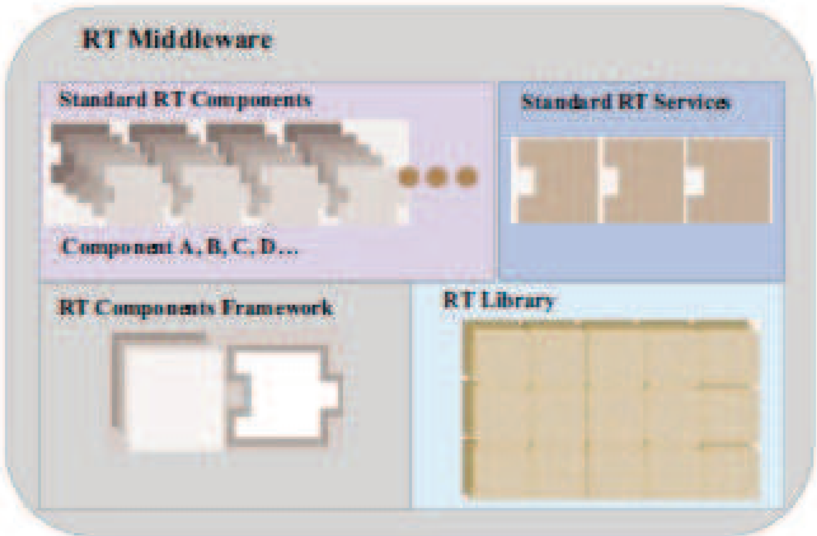

Fig. 1. Structure of Robot Technology Middleware.

2.2 RT Component

RT Component includes RT body object, Inport object and Outport object. RT Component's features are such as it has activity, interface for connecting output and input data stream and administrative function of component-objects. Figure 2 illustrates the structure of RT component.

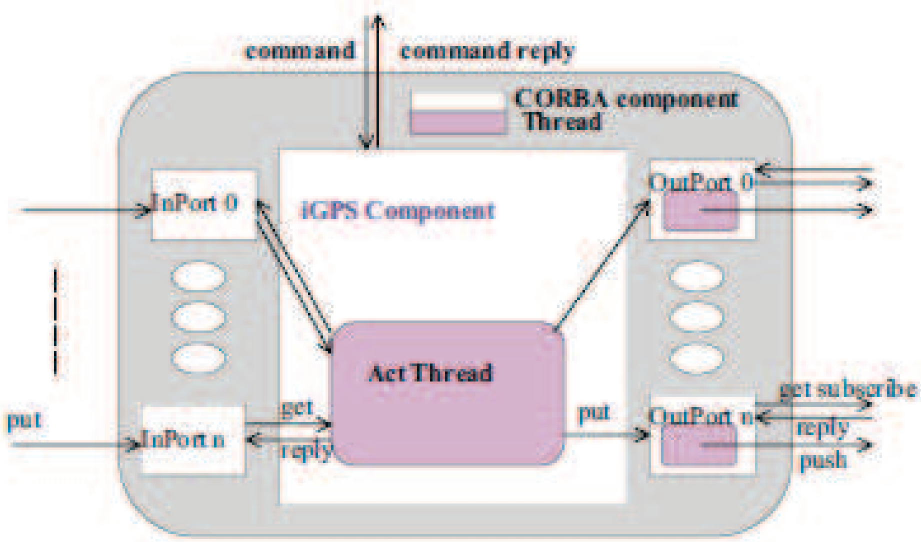

Fig.2. Structure of RT Component.

2.3 RT component manager

Figure 3 illustrates the relationship among RT Component, Manager and name service. RT Component Manager manages all RT Component, controls activity thread, adds or deletes the object of RT Component and does instance of Component. Registering all RT Components' names and object references to name server using CORBA name service enable the other host to access the them in different network.

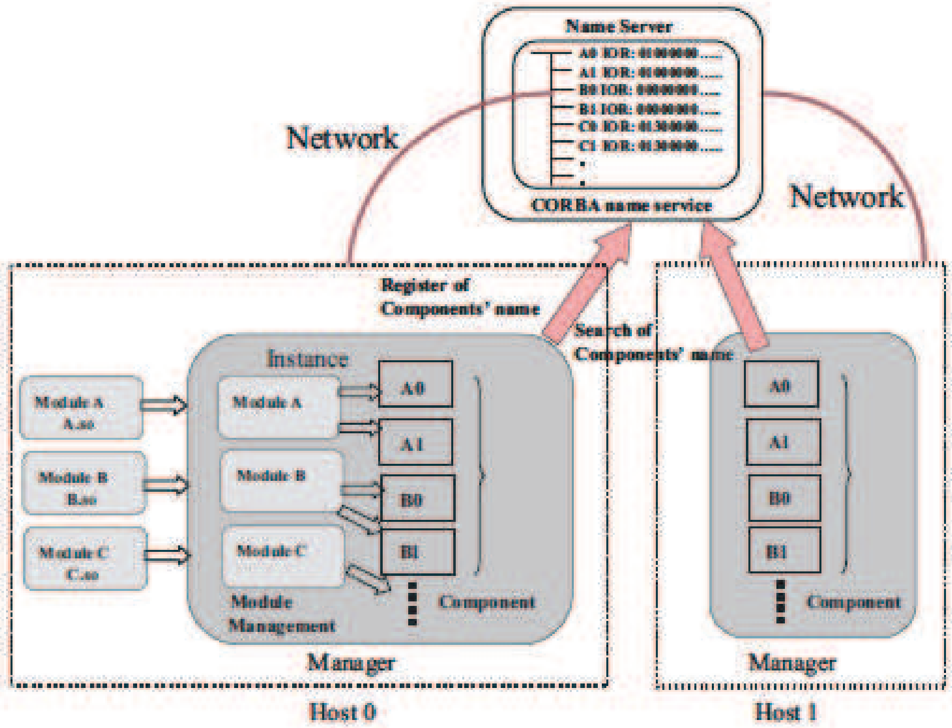

Fig. 3. The relationship among RT Component, Manager and name service.

3. System demonstration at the 2005 World Exposition, Aichi, Japan

Home robot integration system and network monitoring system using QuickCam Orbit cameras were developed and demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan. Figure 4 shows the architecture of the developed intelligent home service robotic system.

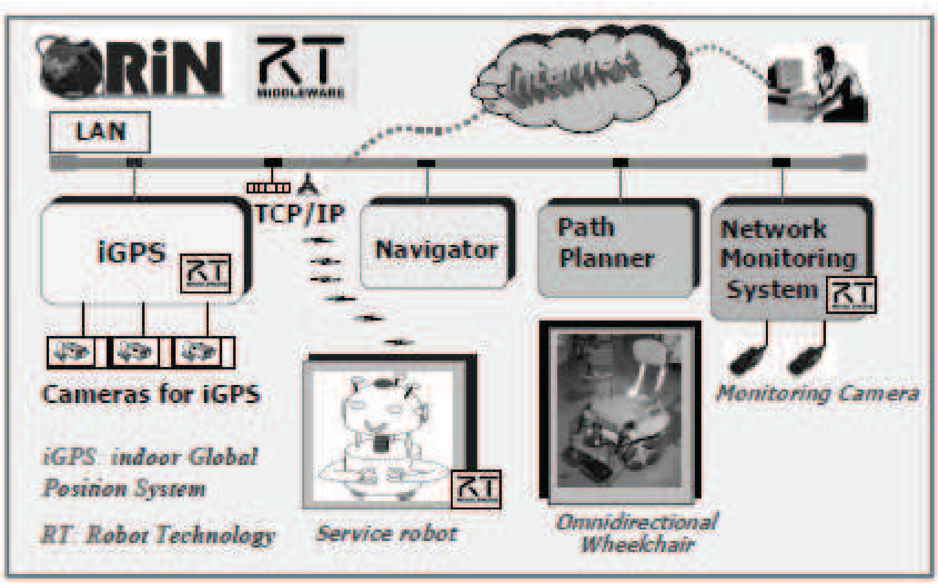

Fig. 4. Structure of the developed intelligent service robotic system.

3.1 Multi-robot systems

The omnidirectional wheelchair was developed to give the aged or disabled the same degree of mobility that healthy people enjoy. The weight of the developed wheelchair is about 80 kg, the maximum acceleration is about 0.41 G, the maximum velocity is about 1.3 m/s and the time of continuous operation time is about 0.5 hour. The user can control wheelchair by a joystick that is applied mainly as user interface. In addition, a novel steering interface was also developed for holonomic omnidirectional power wheelchair to observe user's body action such as tilting an upper body or twisting a waist in order to get user's intention. The observation of changing user's posture on wheelchair is implemented by force sensors arranged under the seat. The user can maneuver the wheelchair by simple body actions with free both hand, so that he or she can enjoy playing a sport such as tennis. Figure 5 illustrates the developed omnidirectional power wheelchair and its specification.

Specifications

Weight	80kg
Maximum acceleration	4.1 m/s^2
Maximum velocity	1.3 m/s
Continuous operating time	0.5 hours

Fig. 5. The developed omnidirectional wheelchair

An intelligent omnidirectional errand robot was developed to help the disabled wheelchair user. The weight of the developed errand robot is about 50 kg, the maximum velocity is about 1.2 m/s and the time of continuous operation time is about 2 hour. When the aged or disabled wheelchair user sends a command such as "juice, please" via PDA (Personal Digital Assistance), the errand robot can autonomously move to the place where the juice is, and take the juice, then confirm the position and orientation of the wheelchair, move to the place where the aged or disabled wheelchair user is, and hand off the juice. Even if the wheelchair user changed the position or orientation while the errand robot was executing a task, the robot can recognize the changes and perform the task autonomously because the robot can get the information of the wheelchair user's position via iGPS. All robots are equipped with infrared (IR) LED units. An IR pass filter, which removes light in the visible spectrum, is attached to the TV camera lens. These TV cameras detect the position and orientation of mobile robot accurately. Figure 6 illustrates the developed errand robot and its specifications.

Specifications

Weight	50kg
Maximum acceleration	0.4 m/s^2
Maximum velocity	1.2 m/s
Continuous operating time	2 hours

Fig. 6. Developed errand robot system.

3.2 Task management server

We developed task management server for home integration robotic system. The task management server accepts commands from the robot operation management server and manages the behavior of each robot. The task management server has two functions. The first function is to manage the behavior of the errand robot. The server accepts the the commands given in Robot Action Command (RAC) format from the robot operation management server. Figure 7 illustrates some samples of Robot Action Commands divided into a sequence of robot behaviors. For example, Robot Action Command “bring a canned drink” can be divided into sequence of robot behavious such as: “go to the place where the canned drink is, pick up the container, go to the place where the wheel chair user is , give the object and so on”. These commands are sequentially transmitted to the navigation server via CORBA communication. When the navigation server notifies the Task Management Server that it finished executing the given command, the next command is transmitted to the navigation server from the task management server.
The second function is to manage the state of the robot. The state of robot is periodically transmitted from the navigator server, and stored in its memory. The task management server was developed by ORiN (Open Resource interface for the Network/Open Robot interface for the Network, OriN”, http://www.orin.jp/) provides integrated interface to access to the devices on the network. One of them is Application Program Interface (API) and the other is communication interface which absorbs the differences between the various specifications of the data storage or the communication protocols between personal computers and devices. Furthermore, ORiN provides protocols to connect to the Internet.

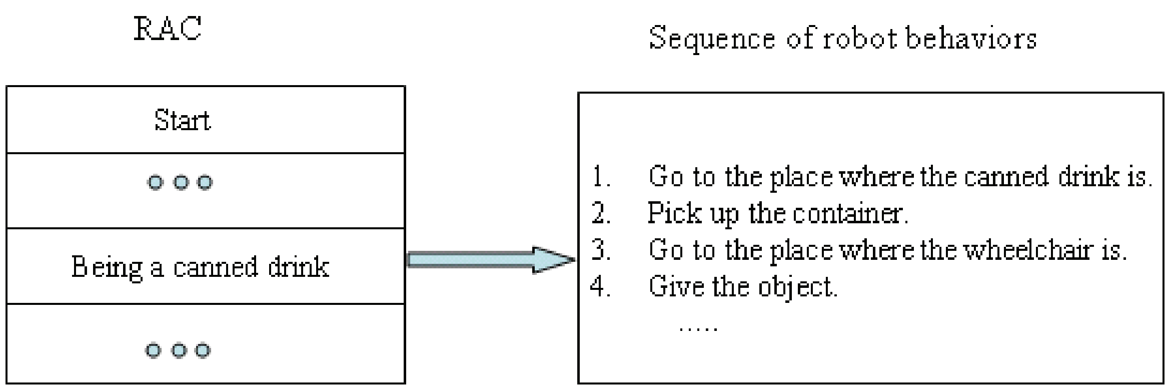

Fig. 7. RAC into a sequence of robot behaviors

3.3 Robot navigation server

The Robot Navigation Server receives the task level directive which was constructed by the Task Management Server, gathers information of robots and environments, generates the operation level instructions in real time, and sends the instructions to all robots. Figure 8 illustrates the architecture of developed Robot Navigation Server.

The navigation server is implemented on a desktop PC (AthlonTM64 3500+) running FreeBSD 4.11-STABLE. This server is designed as a part of the distributed system, and CORBA (implementation of omniORB4) is employed as middleware.

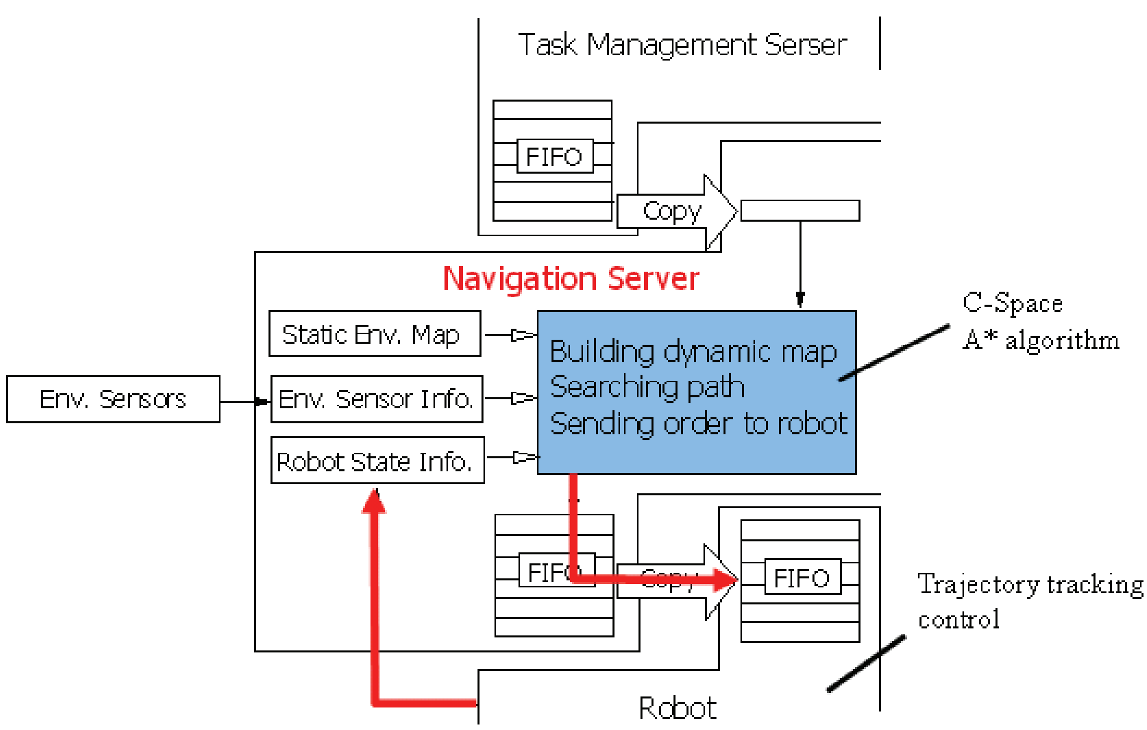

Fig. 8. Developed Errand mobile Robot navigation server.

3.3.1 Task level directive

The following seven task directives are issued from the Task Management Server to the Errand Robot Navigation Server: (1) Canceling a directive, (2) Moving a robot absolutely, (3) Moving a robot relatively, (4) Loading a container, (5) Unloading a container, (6) Pushing a container halfway, and (7) Pulling a container. Directive (2) is used to move the robot toward a target position given with respect to the global coordinate system. When the robot receives Directive (3), it moves to the relatively designated position. Directives (4) and (5) are sent when it is necessary for the robot to load/unload a container. Directives (7) and (8) are issued to have the robot give an object to the user. The navigation server handles only one task level directive per robot at the any time. If two or more directives are received at the same time, the extra directives are rejected. The user can also send directives using the PDA, or another command terminal, manually. If the received directive is impracticability, then the navigation server rejects the directive and informs the task management server that the navigation server cannot executing a directive. When the directive is finished, the navigation server notifies the work management server and then waits for a new directive.

3.3.2 Operation level instructions

The environment map used by the robot navigation server consists of a static map and a dynamic map. The static map includes geometric information, such as the shape of the robot workspace and stationary objects like desks or shelves. The static map is built beforehand. After the robot finishes an operation level instruction, the navigation server gathers information about uncharted dynamic bjects such as other robots and chairs, sensed by the sensors arranged on the robots or in the environment. And the navigation server refreshes the environment maps using this information. Using the update map, the navigation server builds Configuration space (C-Space) as a search space to find a collision free path. The A* algorithm (Gakuhari, et al., 2004) is used to determine the shortest path in the approximate C-Space. The obtained path is represented as a sequence of cells.

One-step motion from one cell to a neighboring cell in the approximate C-Space and manipulation of the container are treated as one operation level instruction. Therefore, one task level directive is divided into a sequence of operation level instructions. The navigation server stores these operation level instructions in a multistage pipeline. The instructions are copied to the multistage pipeline of the robot immediately via CORBA communication. After the robot finishes executing all of the instructions, in other words, when the pipeline is emptied, the navigation server issues the next instructions until the given task level directive will be finished.

As shown in Figure 9, the robot generates a trajectory from instructions that represent translation and rotation motion. The current pose of the robot can be determined by iGPS and an odometer, and a smooth trajectory from the current pose to a target pose can be generated using the third-order B-spine interpolation algorithm.

The navigation server runs on i386 architecture PC (CPU: Pentium 4 2.40B GHz, Memory: PC2100 512 MB) with FreeBSD 4.11-RELEASE. According to the experimental results, the server can dynamically navigate the robots in real time without a collision or a dead-lock in an environment. When the robot finishes one instruction, the robot notifies the navigation server and deletes the corresponding instruction from the pipe-line. When the server received the notification, it also deletes the corresponding instruction from the pipe-line, and checks the number of the instructions remaining in pipe-line. If the number is less than the configured value, then the navigation server generates new instructions.

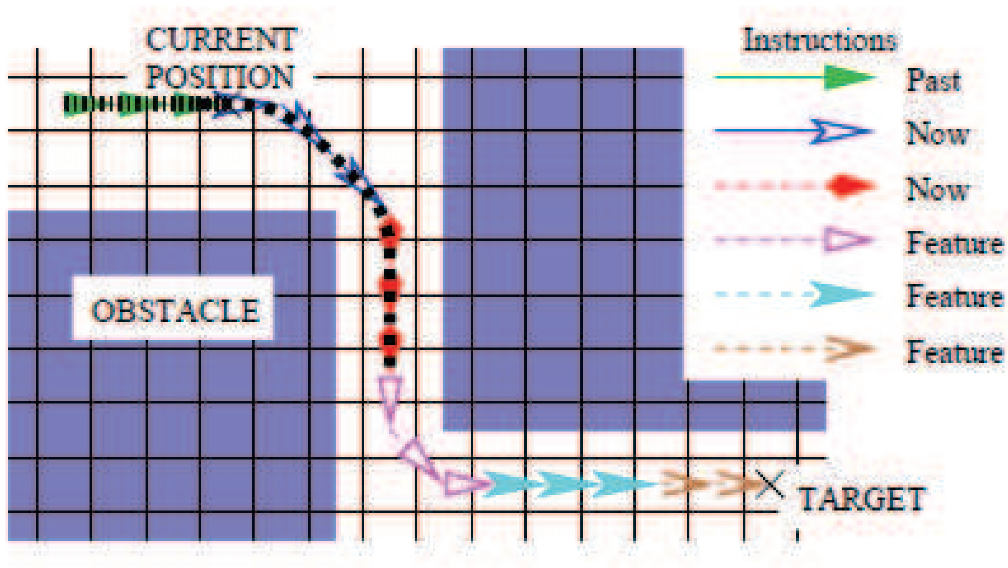

Fig. 9. Trajectory generation from s sequence of the task level instructions.

3.3.3 Dealing with abnormal circumstances

When the robot enters an abnormal state, all servers and the robot process stop immediately. The typical examples of abnormal circumstances are given as following:

- When the user gives the stop directive by using a PDA or something commanding terminal, the task management server gives the stop directive to the robot navigation server. When the navigation server receives the stop directive, it sends the cancel instruction to the robot, and removes all of the operation level instruction from the pipeline. The navigation server then waits for the next directive from the task management server.
- When the emergency button of the robot is pushed, the robot stops moving and notifies the navigation server. The navigation server then dequeues all of the operation level instructions from the pipeline. The navigation server then waits for the next directive from the task management server.
- If disconnection of the communication channel between the navigation server and the robot is detected, the navigation server and the robot try to recover the connection. If the disconnection is fatal, autonomous recovery is impossible and the connection must be fixed manually.

3.4 iGPS RT functional component

We developed iGPS RT functional component in order to enable system integration easier. An indoor Global Positioning System (iGPS) has been developed to localize the omnidirectional mobile robot. IEEE 1394 cameras, are mounted on the ceiling so that the cameras overlook the robot's moving area (Hada at el., 2005). We evaluated accuracy of iGPS by experiments. We selected 24 points distributed in the Lab about 7000 mm×7000 mm area, and measured them using iGPS and the maximum value of measurement error is 38 mm. This result verified that the accuracy of iGPS is enough for navigation of mobile robot.

4. Network distributed monitoring system using QuickCam Orbit cameras

In order to enable a remote user to get a better understanding of the local environment, media streams must be received and transmitted in real-time in order to improve interaction in home integration robot system. We implemented video/audio RT component based on RT Middleware, and OmniCORBA IIOP is employed as message communication protocol between RT component and requester. The QuickCam Orbit (Logitech Co.) cameras were used in our system with high-quality videos at true CCD 640×480 resolution, automatic face-tracking and mechanical Pan, Tilt and face tracking feature. This camera has a maximum video frame rate is 30 fps (frames per second) and works with both USB 2.0 and 1.1. The area of the booth used to demonstrate the developed robotic system was approximately 4.5×5 m^2, so two cameras were set up in the environment. The cameras were able to view the area in which the omnidirectional wheelchair and errand robot move by adjusting the mechanical Pan and Tilt of the cameras. The structure of the developed RT video stream functional component is shown in Figure 10. This RT component has one Inport for camera property setting and Outport 1 for video data and Outport 2 for status data of camera control.

- Inport: camera property for camera's setting
- Outport1: video data
- Outport2: status data for camera control

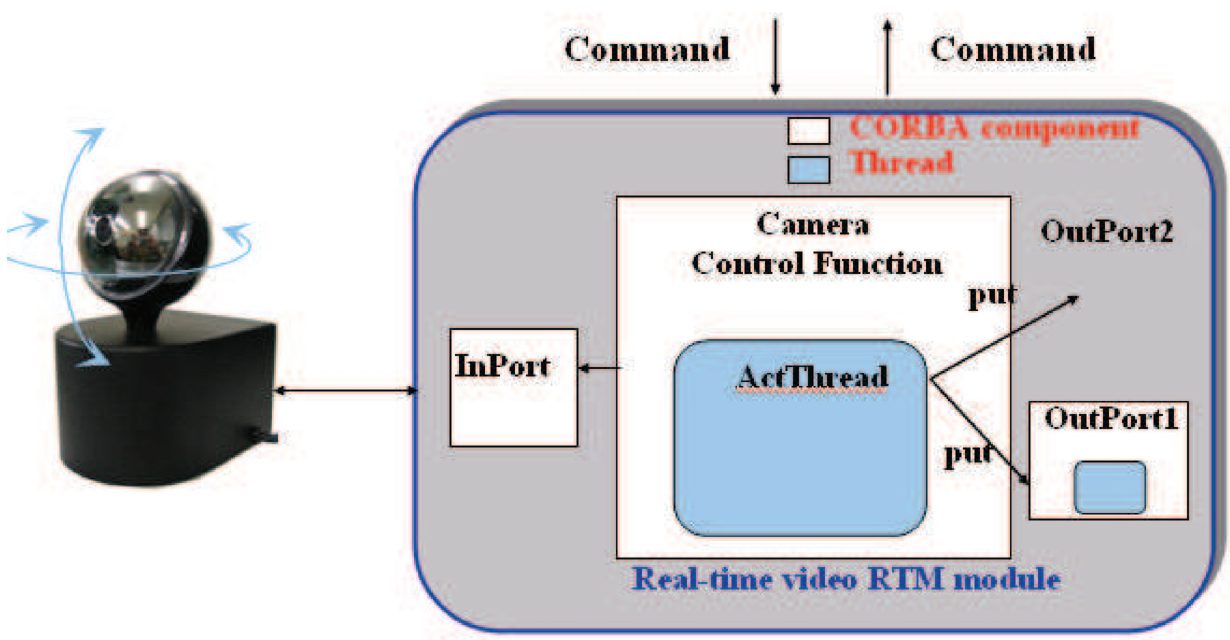

Fig. 10. Video/audio RT component developed based on RTM.

Figure 11 illustrates the class structure of the developed video RT component. The camera's control function classes includes:

- RtcBase: OpenRTM-aist-0.2.0 component base class.
- InPortBase: OpenRTM-aist-0.2.0 InPort base class.
- OutPortBase: OpenRTM-aist-0.2.0 OutPort base class.

- InPortAny<TimedUShortSeq>: InPort template class.
- OutPortAny<TimedUShortSeq>: OutPort template class.
- RtcManager: RT component management class.
- CameraRTC: camera control RT component.
- CameraComp: camera control RT component main class.
- CameraControl: camera operation class.

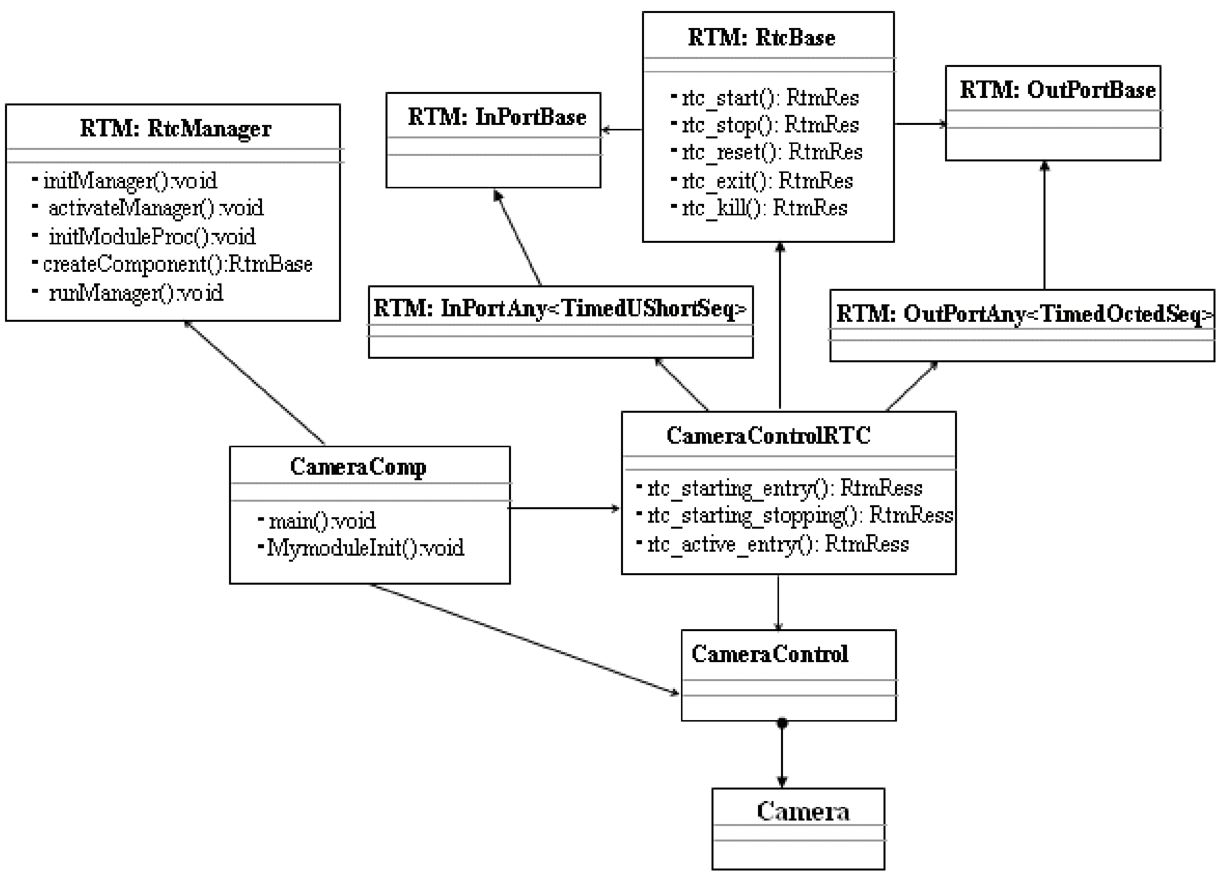

Fig. 11. Class structure of the developed RT component.

In addition, we developed a graphic user interface (GUI) for the video stream system that provides a remote video stream camera zoom and pan-tilt adjustment, and a chat function that allows a remote user to communicate with a local user. When the user sends a request for video, the system will autonomously display the GUI. The user can click “Connect” and input the IP address of the computer on which the RT video component is running to view a real-time video feed.

The RT video stream component was implemented by Visual C++, Microsoft visual studio.net 2003. A performance test of the developed real-time video stream was conducted to examine the possibility of using a live video feed to monitor the state of the elderly or disabled wheelchair user. The video server is run on Windows 2000 Professional (1.9 GHz, Pentium4), and the video client is run on Windows XP (2.4 GHz, Pentium4). The average frame rate is approximately 16.5 fps (video format 320×288). Figure 12 illustrates the architecture of the developed network monitoring system based on RTM.

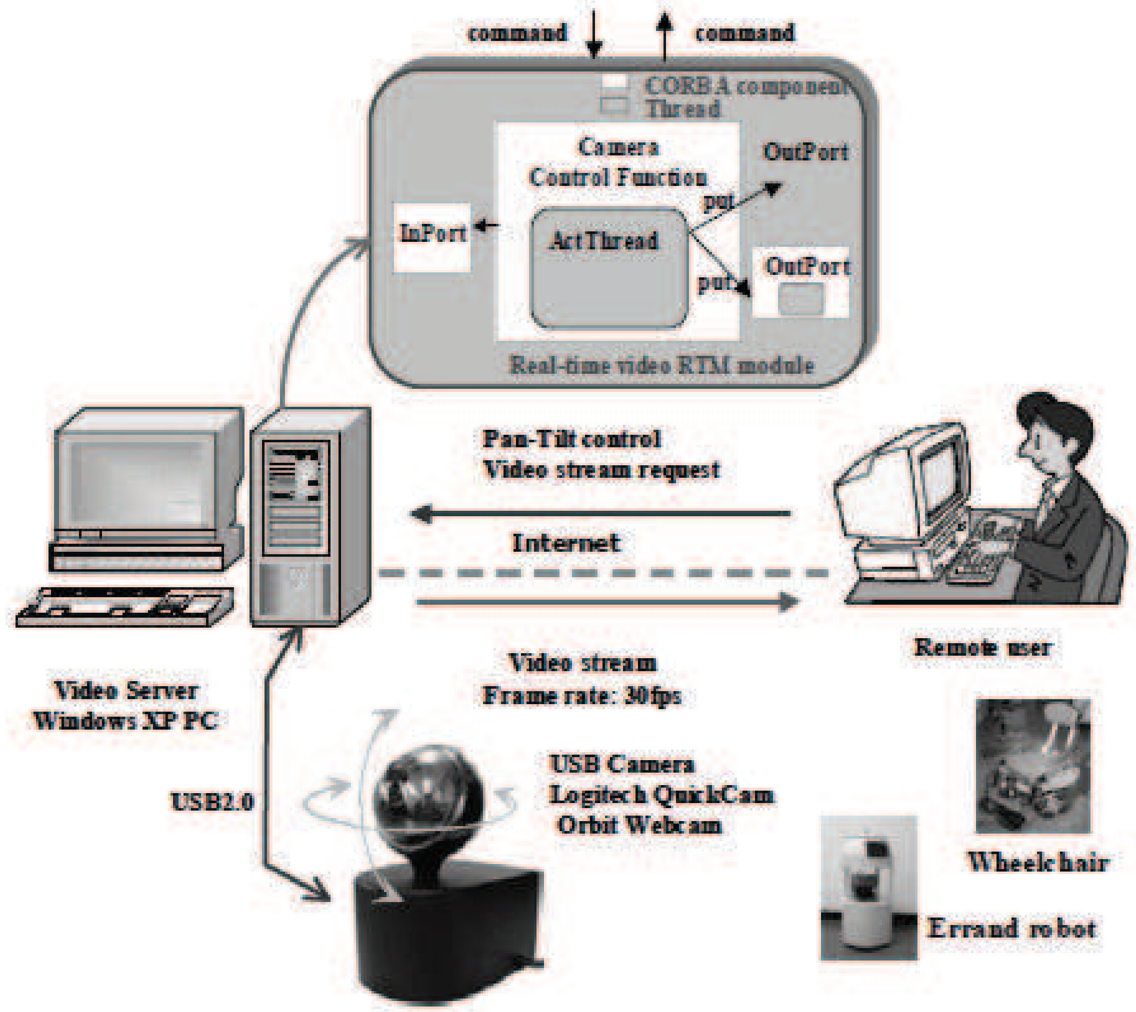

Fig. 12. Structure of RT video stream functional component.

5. Experimental results

Home integration robotic system was demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan. Figure 13 illustrates the scenery of demonstration in the 2005 World Exposition, Aichi, Japan. Figure 13(a) is a modelled living room at the prototype robot exhibition and 13(b) is the booth for our developed system demonstration. Figure 13(c)-(f) illustrates some images of task performance demonstration of robotic system performing a service task. The wheelchair user can issue an order to the robot to bring objects such a canned drink via PDA. Then the errand robot starts to move toward the front of the shelf where the container holding the target canned drink is placed and loads the container. The errand robot can offer the canned drink to the wheelchair user because the robot can obtain position information of the wheelchair via iGPS. Even if the wheelchair user changed the position or orientation while the errand robot was executing a task, the robot can recognize the changes and perform the task autonomously. Fig. 13(g)-(i) illustrates the video stream for monitoring the state of robotic systems working. The developed network distributed monitoring system can monitor the state of robotic system's working and the state of the aged or disabled in demonstration. Cameras for monitoring the environment were connected to the computer running on Windows XP (2.4 GHz, Pentium4), and GUI is run on the other same specification Windows XP (2.4 GHz, Pentium4). Two computers are connected in a LAN. The average frame rate is approximately 18.5 fps. Figure 14(a)-(h) shows the performance demonstration of the omnidirectional powered wheelchair. The user operates the wheelchair through the joystick skilfully (Figure 14(a)-(d)). The user can also operate the wheelchair via a body action control interface which enables hands-free maneuvering of the wheelchair, so that he or she can enjoy playing a ball with two hands (Figure 14(e)-(h)). The demonstration time was approximately held twice a day. A total of 22 demonstrations were performed and the errand robot failed to execute its task three times. The success rate is about 86%. The cause of the failure was that the angle of Camera 2 changed over time so that the calibrated camera parameters differed from the original parameters, causing an error in the measurement of the robot position. When Camera 2 was neglected, the robot did not fail to execute its task. The demonstration verified that the developed system can support the aged or disabled to a certain degree in daily life.

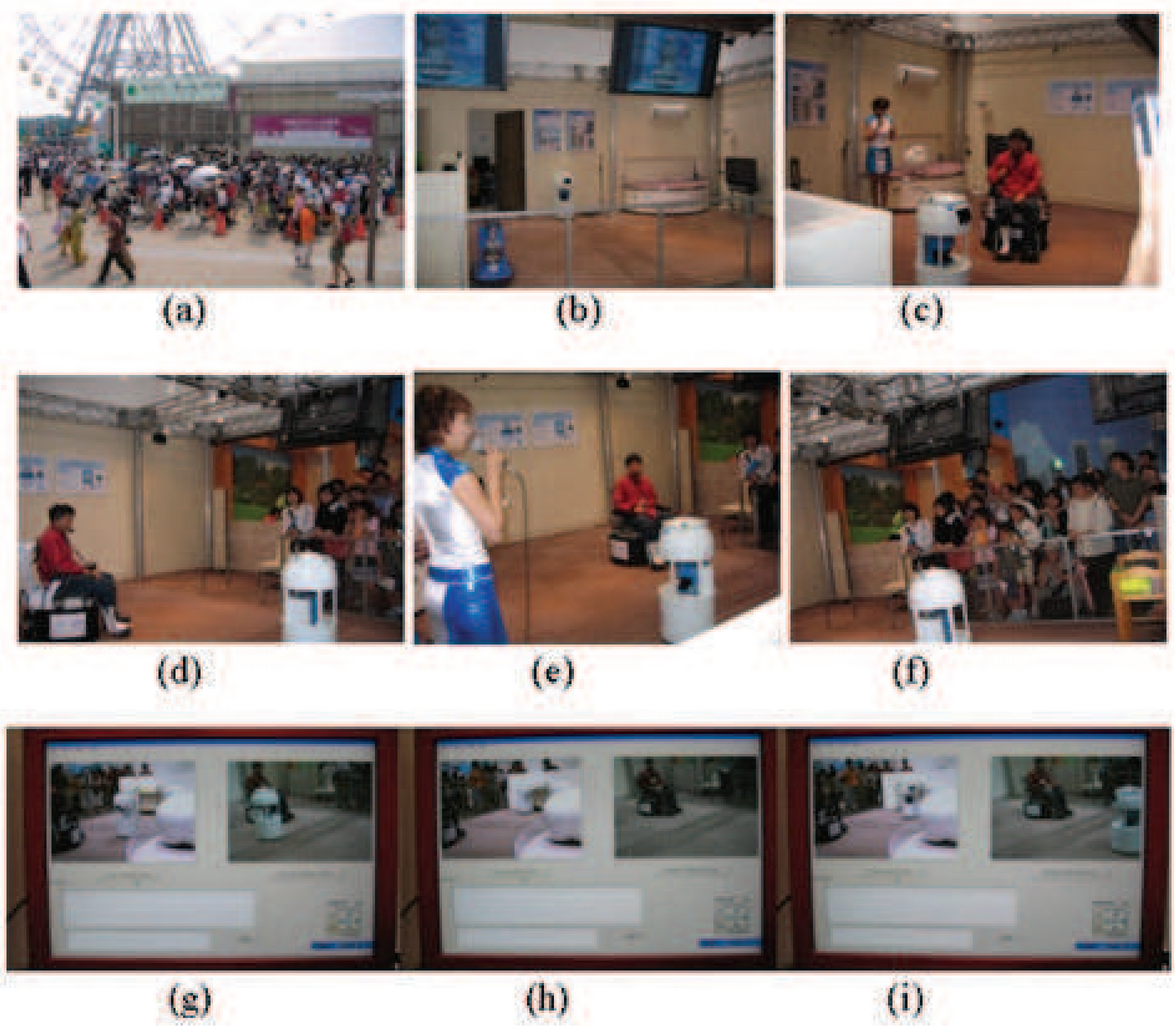

Fig. 13. Some images of task performance demonstration of robotic system performing the task

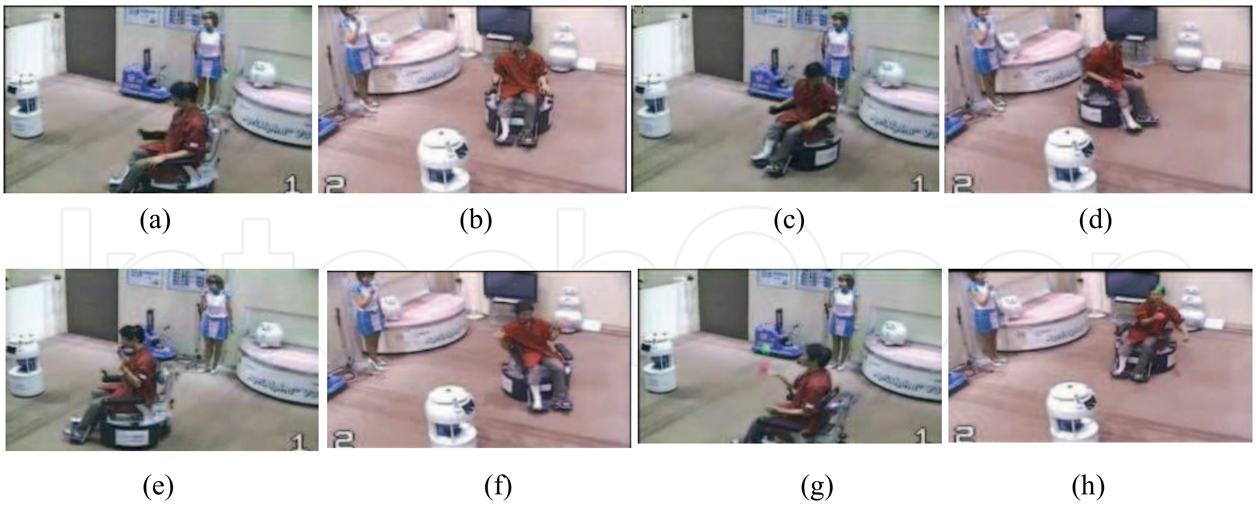

Fig. 14. The demonstration of the omnidirectional powered wheelchair.

6. Conclusion

This paper presented the developed service robotic system supporting elderly or disabled wheelchair users. Home integration system was demonstrated at the prototype robot exhibition from June 9 to June 19 June at the 2005 World Exposition, Aichi, Japan. We developed an omnidirectional wheelchair and its maneuvering system to enable skilful operation by disabled wheelchair user. Since the user can maneuver the wheelchair intuitively by simple body actions and with both hands free, they are able to enjoy activities such as tennis. We also developed an errand robot that can deliver objects such as newspaper or canned drink to disabled wheelchair users. Even if the wheelchair user changed the position or orientation while the errand robot was executing a task, the robot can recognize the changes and perform the task autonomously because the robot can get the information of the wheelchair user's position via iGPS. Network monitoring system using QuickCam Orbit cameras was implemented to monitor the state of robotic systems working.
Because Robot Technology Middleware (RTM) was used in the developed system, we can develop the functional module as RT component, which makes the system has high scaling and inter-operating ability, facilitating network-distributed software and sharing, and makes application and system integration easier. It is also very easy for the user to create new application system by re-using existing RT components, thus lowers the cost of development of new robotic system. For future work, we will develop the other functional robot system components as RT components such as RFID RT object recognition component for object recognition or RT localization component for localizing mobile robot in order to improve the flexibility of the home integration robotic system.

7. Acknowledgements

The home integration robotic system and network distributed monitoring system demonstrated at the prototype robot exhibition from June 9 to June 19 June at the 2005

World Exposition, Aichi, Japan was developed with funding by the New Energy and Industrial Technology Development Organization (NEDO) of Japan. The authors would like to thank System Engineering Consultants (SEC) CO. LTD. for their support in developing system.

8. References

Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W., 2004, Implementation of RT composite components and a component manager, The 22nd Annual conference of the Robotic Society of Japan, IC26.

Kitagaki, K., Suehiro, N., Ando, N., Kotoku, T., and Yoon, W., 2004, GUI components for system development using RT components," The 22nd Annual conference of the Robotic Society of Japan, IC23,2004.

Jia, S., and Takase, K., 2001, Internet-based robotic system using CORBA as communication architecture, Journal of Intelligent and Robotic System, 34(2), pp. 121-134, 2001.

Jia, S., Hada, Y. and Takase, K., 2004, Distributed Telerobotics System Based on Common Object Request Broker Architecture, The International Journal of Intelligent and Robotic Systems, No.39, pp. 89-103, 2004.

Gakuhari, H., Jia, S., Hada, Y., and Takase, K., 2004, Real-Time Navigation for Multiple Mobile Robots in a Dynamic Environment, Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, pp. 113-118.

Hada, Y., Gakuhari, H., Takase, K.and Hemeldan, E.I., 2004, Delivery Service Robot Using Distributed Acquisition, Actuators and Intelligence, Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robots and System (IROS'2004), pp. 2997-3002.

Hada, Y., Jia, S., Takase, K., Gakuhari, H., Nakamoto, H., and Ohnishi, T., 2005, Development of Home Robot Integration System Based on Robot Technology Middleware, The 36th International Symposium on Robotics (ISR 2005), TU4H6, Japan.

Message-orientated middleware: http://sims.berkeley.edu/ courses/is206/f97/ GroupB /mom/.

Ohnishi, T., and Takase, K., 2002, The maneuvering system of omnidirectional wheelchair by changing user's posture, Proceeding of 2002 international Conference on Control, Automation and System (ICCAS2002), pp. 1438-1443, 2002. http://www.orin.jp/.

Object Management Group, http://www.omg.org.

Object Oriented Concepts, Inc., http://www.omg.org.

Java remote method invocation:

http://java.sun.com/products/jdk/rmi/index.html.

http://www.is.aist.go.jp/rt/.

Nagi, N. Newman, W.S., Liberatore, V. (2002), An experiment in Internet-based, human-assisted robotics, *Proc. of IEEE Int. Conference on Robotics and Automation (ICRA'2002)*, Washington, DC, USA, pp.2190-2195.

Schulz, D., Burgard, W., Fox, D. et al.: (2002), Web Interface for Mobile Robots in Public Places, *IEEE Robotics and Automation Magazine*, 7(1), pp. 48-56.

Stein, M. R. Stein, (2000), Interactive Internet Artistry, *IEEE Robotics and Automation Magazine*, 7(1) (2000), pp. 28-32.

Advances in Service Robotics

Edited by Ho Seok Ahn

ISBN 978-953-7619-02-2
Hard cover, 342 pages
Publisher InTech
Published online 01, July, 2008
Published in print edition July, 2008

This book consists of 18 chapters about current research results of service robots. Topics covered include various kinds of service robots, development environments, architectures of service robots, Human-Robot Interaction, networks of service robots and basic researches such as SLAM, sensor network, etc. This book has some examples of the research activities on Service Robotics going on around the globe, but many chapters in this book concern advanced research on this area and cover interesting topics. Therefore I hope that all who read this book will find lots of helpful information and be interested in Service Robotics. I am really appreciative of all authors who have invested a great deal of time to write such interesting and high quality chapters.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821