2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

A RTM-Based Home Service Robots Monitoring System

Songmin Jia and Kunikatsu Takase

Abstract— **In this paper, a RTM-based network distributed monitoring system for human assistance robot system was developed to improve the interaction among the users and local service robotic system. Home integration robot 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. Improvements of network distributed monitoring system using IEEE1394 cameras with high performance and high resolution have been done in order to extend the application of system. Robot Technology Middleware (RTM) was used in the developed system. By using RTM, we can develop cameras functional elements as "RT software components" that can be implemented by different programming languages, run in different operating system, or connected in different networks to inter-operate. It is also easy to create comprehensive robot system application by re-using existing modules thus facilitating network-distributed software sharing and improving the cost of writing and maintaining software.**

I. INTRODUCTION

Many service robotic systems have been developed in order to improve care cost and the QoL of the elderly people in the population-aging society. Our HARSP (Human-Assistance Robotic System Project) project has been developing a network distributed Human-Assistance Robotic System since 2000 [4]. 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 has been doing from practical viewpoint. In order to enable a remote users to get a better understanding of the state of the robots working and the state of the aged or disabled, we developed network distributed monitoring system to transmit media streams in real time to improve interaction between a remote user and local robot system. 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. Robot Technology Middleware (RTM) was used in the developed system. RT Component has standard interfaces and hides its implementation, and has network transparârency and efficient communication channel. RT Components of the system can be implemented by different programming languages, run in different operating system, or connected in different networks to inter-operate. Baba, T. et al developed Physical Agent System (PAS) that supports daily activity of human using RTM [3]. Inamura, W. et al introduces how to realize real-time control of system by applying the Composite Component framework [6]. We develop robotic functional elements as "RT software component", which makes it easy to create new robot system by re-using existing modules and to lower the cost of development of new robot application. In order to enhance the performance of the network distributed monitoring system, we developed IEEE-1394 cameras RT control components in order to make application and system integration easier. For different vendors of IEEE-1394 cameras, system supplies classes method easy to implement them as RT components. Being different from QuickCam Orbit cameras, IEEE-1394 cameras were widely used in robot field and they have high performance and high resolution. This paper introduces the architecture of the developed system and the results of experiments.

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 and network monitoring system using QuickCam Orbit cameras. Section 4 details the architecture of the enhanced network distributed monitoring system using IEEE-1394 cameras. The experimental results are given in Section 5. Section 6 concludes the paper.

II. ROBOT TECHNOLOGY MIDDLEWARE

Robot Technology Middleware (RTM, [1]) 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 enable system to be language independence, operating system independence, RTM has been developed based on CORBA. CORBA (Common Object Request Broker Architecture) is a distributed computing technologies, there are the others like RMI (Remote Method Invocation, [9]), DCOM (Distributed Component Object Model), MOM (Messages Oriented Middleware). In contrast to all of these, CORBA uses an Object Request Broker (ORB, [7]) 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

S. Jia is with Graduate School of Information System, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-City, Tokyo 182-8585, Japan jia@taka.is.uec.ac.jp

K. Takase is with Graduate School of Information System, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-City, Tokyo 182-8585, Japan takase@taka.is.uec.ac.jp

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).

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, [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.

3) RT Component Manager: Figure 1 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.

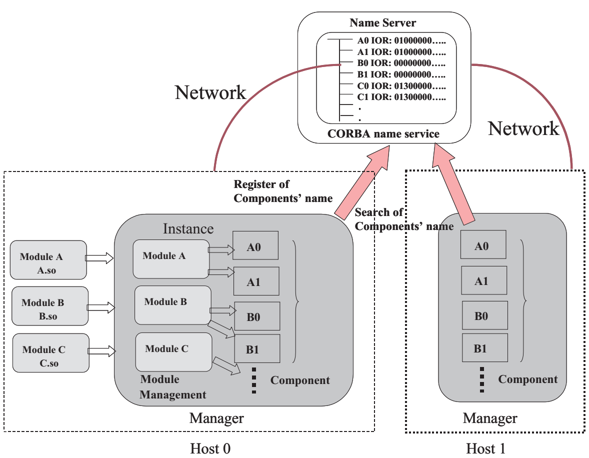


Fig. 1. Structure of RT Component Manager.

III. System Demonstration at the 2005 World Exposition, Aichi, Japan

Home robot integration system and network monitoring system using QuickCam Orbit cameras were developed and were demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan. Figure 2 shows the architecture of the developed system.

The omnidirectional wheelchair and maneuvering system have been developed to enable the disable person to operate it skillfully. We also developed the omnidirectional intelligent errand robot which can deliver the necessary objects such as drink and newspaper to help the aged or disabled who is taking the wheelchair. The Task Management Server was

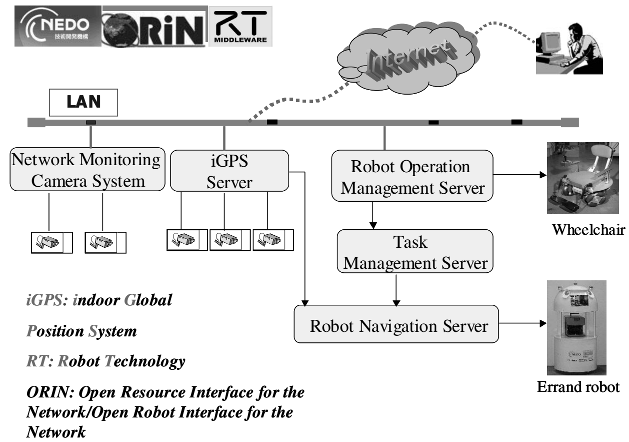


Fig. 2. Architecture of the developed robotic system.

developed to manage the behavior of each robot. iGPS was used to localize the omnidirectional wheelchair and errand robot in the moving area [5]. Robotic Navigation Server ware also developed to navigate the errand robot how to move from the start position to goal in order to finish a task-level service task. Robot Navigation Server receives the task level commands from task management server, gathers information about the robots and environment, and plans to find optimal path for the mobile robots to move.

In order to enable a remote user to get a better understanding of the local environment of mobile robot working, it is necessary to receive and transmit media streams in real time to improve interaction in network distributed service robotic system. Network monitoring system using QuickCam Orbit cameras was developed to visualize home robot integration system working, the state of the age and disabled, and to enable the remote user to know what is going on in local environment.

The QuickCam Orbit cameras were used in our system with automatic face-tracking and mechanical Pan, Tilt and face tracking feature. It mechanically and automatically turns left and right for almost a 180-degree horizontal view or up and down for almost 90 degree top-to-bottom view. Its maximum video frame rate is 30 fps (frames per second) and works with both USB 2.0 and 1.1.

We implemented video stream RT component based on RT Middleware, and OmniCORBA IIOP is employed as message communication protocol between RT component and requester.

Booth for demonstration of our developed robotic system is approximately $4.5{\times}5m^2$ at the 2005 World Exposition, Aichi, Japan. So two QuickCam Orbit cameras were set up in the environment, and they can overlook the area the omnidirectional wheelchair and errand robot moves in by adjusting the mechanical Pan and Tilt of the cameras. The structure of the developed RT video stream monitoring system using QuickCam Orbit cameras was shown in Figure 3.

In addition, we developed a graphic user interface (GUI) for the video stream system that provides a remote video stream a 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.

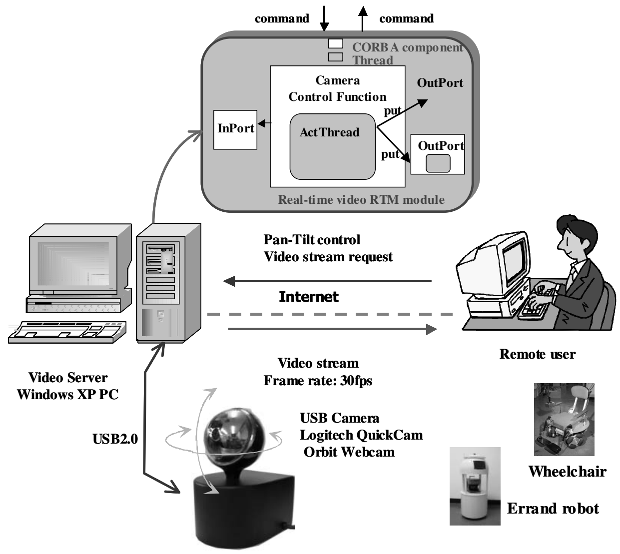


Fig. 3. Structure of RT video stream functional component.

The RT video stream component was implemented by the 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 robotic system. The video server is run on Windows 2000 Professional (1.9GHz, Pentium4), and the video client is run on Windows XP (2.4GHz, Pentium4). The average frame rate is approximately 16.5fps (video format 320×288).

IV. Performance Enhancement of Network Distributed Monitoring System

In order to enhance the performance of the network distributed monitoring system, we developed IEEE-1394 cameras RT control components in order to make application and system integration easier. For different vendors of IEEE-1394 camera, system supplies method easy to implement RT component. A new robot system application can re-use existing modules thus lowering the cost of development. The developed IEEE-1394 cameras RT control components allow other applications in different network to communicate directly with them. Being different from QuickCam Orbit cameras, IEEE-1394 cameras were widely used in robot field and they have high performance and high resolution. Figure 4 illustrates the configuration of the enhanced network distributed monitoring system.

A desktop personal computer running the Windows XP operating system (2.4GHz, Pentium4) with IEEE-1394 buses

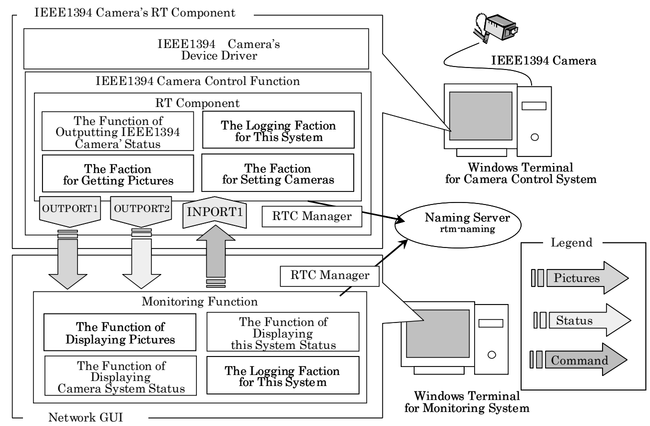


Fig. 4. Structure of improved network distributed monitoring system.

is used in the system for cameras control. IEEE-1394 cameras (Flea, Point Gray Research Inc.) as examples (widely used in robot field) were used in the network monitoring system to take video of the environment and multi-cameras can connected to one computer to be controlled. IEEE-1394 cameras' RT components (RTM-0.2.0) were implemented. The user can control them to start to take video or to set up the configuration of cameras according to the commands from network GUI.

Network GUI was also developed and was run on the other computer (Windows XP operating system 2.4GHz, Pentium4) to enable the user to operate the real-time video stream from IEEE-1394 cameras. User can start, stop the IEEE-1394 cameras' RT components thus starting or stopping the display of video stream and setting up the configuration of cameras. Common Object Request Broker Architecture (omniCORBA 4.0.5) is employed as communication architecture between the network GUI and camera control computer. The sequence of the developed system is shown in Figure 5.

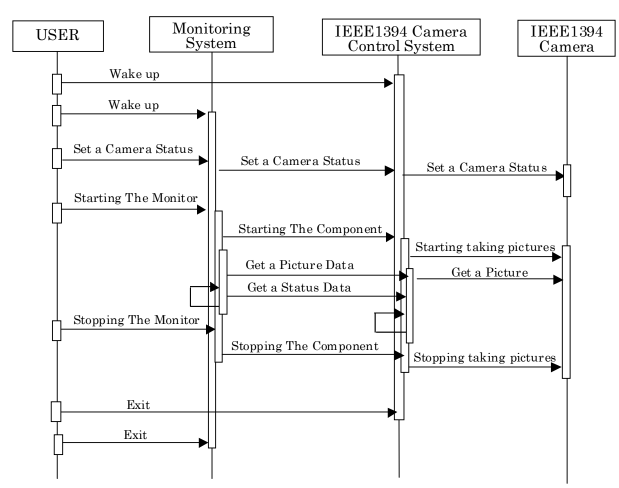


Fig. 5. Sequence of the developed network distributed monitoring system.

A. IEEE-1394 Camera

The Flea color camera were selected as example in the developed system because it was widely used in robot field. The Flea color camera is Point Grey's most compact IEEE-1394 digital camera, designed to fit spaces as small as 30×31mm. With a 1/3" Sony CCD and a 12-bit analog to digital converter, the camera delivers high quality images ideal for demanding imaging applications. It has features like: high resolution with 640×480 or 1024×768; faster standard frame rates 640×480 at 60fps or 1024×768 at 30fps; multiple cameras on the same 1394 bus are automatically synchronized.

B. IEEE-1394 Camera Control RT Component

Figure 6 illustrates the structure of the IEEE-1394 camera control RT component. It has:

- Inport: set the camera's property.
- Outport1: video data.
- Outport2: status data of IEEE-1394 camera control.

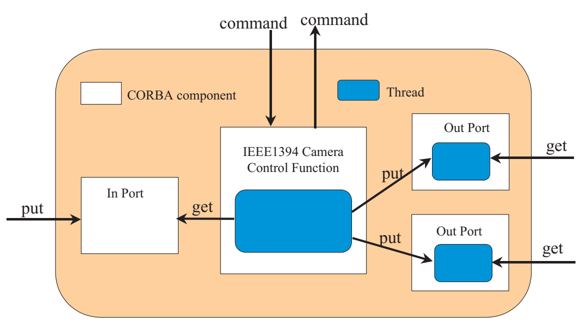


Fig. 6. IEEE-1394 camera control RT functional component.

IEEE-1394 camera control function includes:

- IEEE-1394 camera "image get" function: get the image the user selected by IEEE-1394 camera and change the format (RGB) to display in Windows, output it to OutPort1 of RT component.
- IEEE-1394 camera "set property" function: set the camera's property such as size, frame rate, and mode of the image selected by the user.
- IEEE-1394 camera "output status" function: output the status of camera to OutPort2 of the RT component, inform the user this information via GUI. IEEE-1394 camera status includes IEEE-1394 camera's property, IEEE-1394 camera's initialization success or failure, and IEEE-1394 camera control error.

RT component transition process is shown in Figure 7. It was developed based on RT middleware (RTM 0.2.0). RT component transition process includes:

- rtc_starting_entry(): IEEE-1394 camera initialization/starting record control using IEEE-1394 camera's drivers, setting property of camera according to the command from network GUI.
- rtc_stopping_entry(): stop record control using IEEE-1394 camera's drivers.

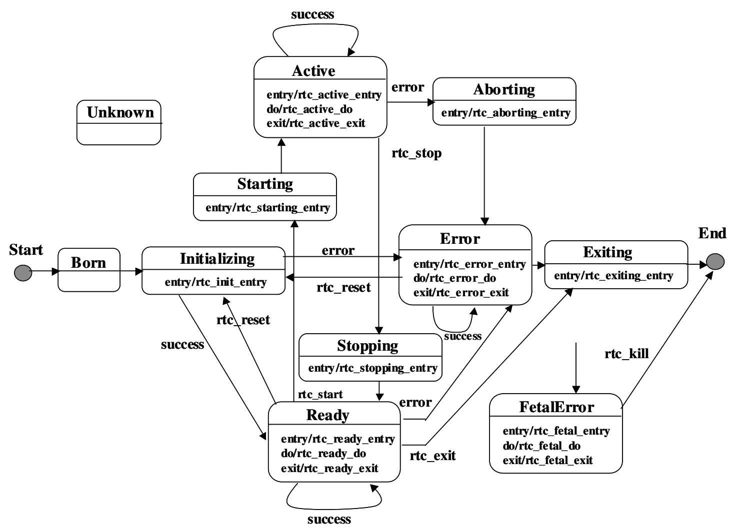


Fig. 7. IEEE-1394 camera RT component transition.

- rtc_active_do(): take the image by IEEE-1394 camera, covert the format of the image data for displaying on the network GUI and output the data to OutPort of RT component.

Figure 8 illustrates the structure of the classes of IEEE 1394 camera control RT component. The IEEE-1394 camera's control function classes includes:

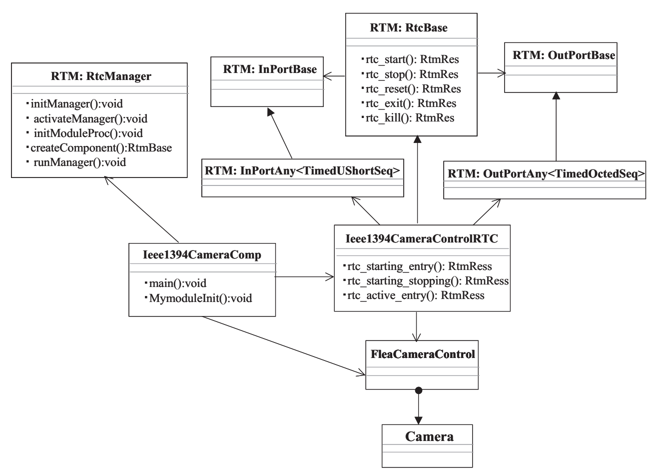
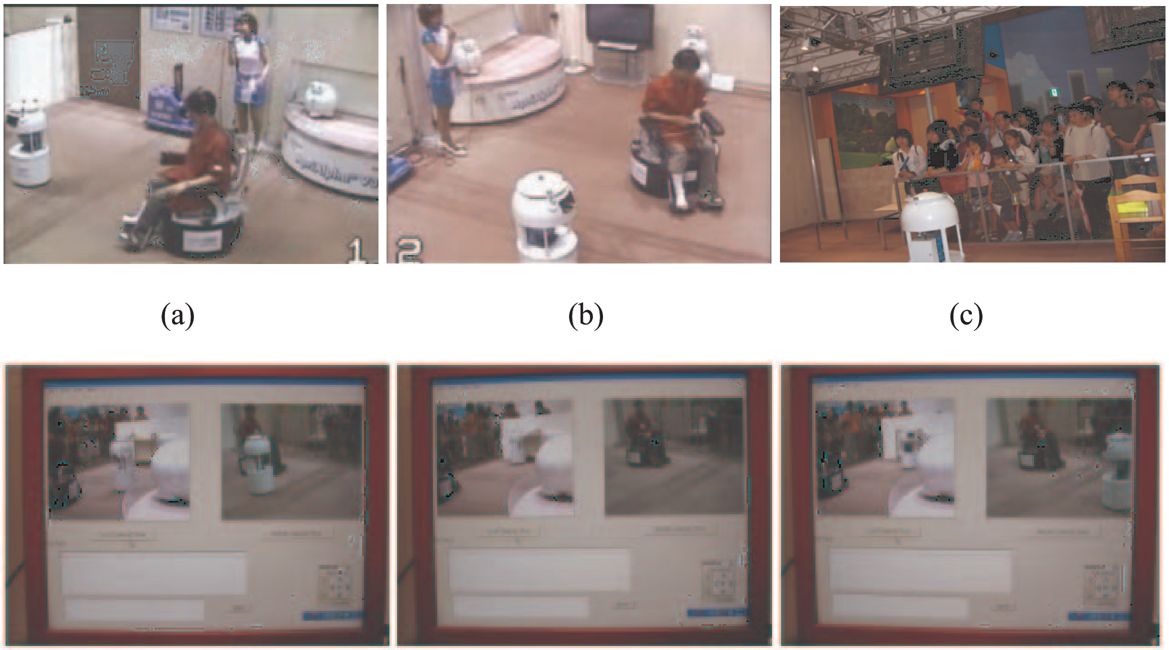


Fig. 8. Structure of classes of IEEE-1394 camera control RT component.

- 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.
- IEEE1394CameraRTC: IEEE1394 camera control RT component.
- IEEE1394CameraComp: IEEE1394 camera control RT component main class.
- FleaCameraControl: IEEE1394 camera operation class.



(a) (b) (c)

(d) (e) (f)

Fig. 10. System demonstration at the 2005 World Exposition, Aichi, Japan.

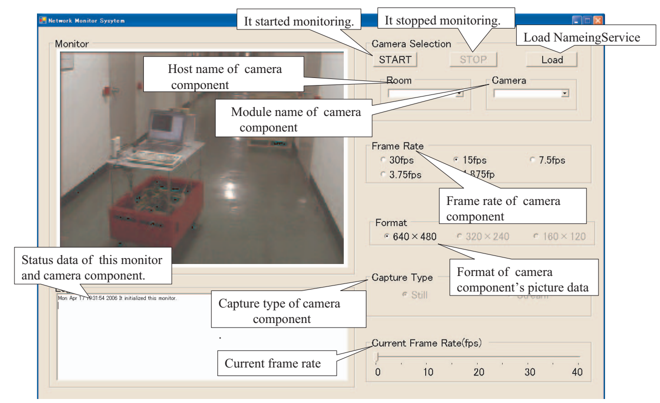


Fig. 9. GUI for network distributed monitoring system with IEEE-1394 cameras.

C. GUI for Network Distributed Monitoring System

Graphic User Interface (GUI, see Figure 9) was designed for the network distributed monitoring system that provides a remote video stream, camera selection, capture type (static or stream), frame rate (30fps, 15fps, 7.5fps and so on) of camera component, format (640×480, 320×240, 160×120) of camera component's image data. The status data such as the starting time which camera was connected or the other message of the system was displayed in log part. GUI also displays the current frame rate (from 0fps to 40fps) of video stream. User pushes "Load Button", selects a room name, camera name, a frame rate and a format the user hopes, then pushes "Start Button", the video stream the user selected will start. When user wants to stop display or to change the camera, he or she just pushes "Stop Button", and selects the other camera's name, the system will stop the video stream transition of the camera now and start the new camera video show the user selected. Because we implemented IEEE1394 cameras as RT components and registered them to naming server using CORBA name service in advance, the user can access any cameras to get video stream even cameras are connected in the different computer or in different network.

V. Experimental Results

We developed a RTM-based network distributed monitoring system for human assistance robot system to improve the interaction among the users and local service robotic system. Home integration robotic system and monitoring system were demonstrated from June 9 to June 19 at the 2005 World Exposition, Aichi, Japan. The user operates the ommidirectional holonomic through the joystick control or via body action control interface that enables hands-free maneuvering of it. The ommidirectional holonomic wheelchair can move in any direction without rotating and enables small adjustments. The wheelchair user uses PDA to issue an order to the errand robot to bring a canned drink or newspaper. Figure 10 shows the scenery of demonstration at the 2005 World Exposition, Aichi, Japan. Figure 10(a)-(c) are the booth for our developed system demonstration and the images of demonstration of the developed home service robotic system. Figure 10(d)-(f) show some images of the developed network monitoring system in demonstration using QuickCam Orbit cameras in order to know the state of robotic system's working and the state of the aged or disabled. A total of 22 demonstrations were performed and

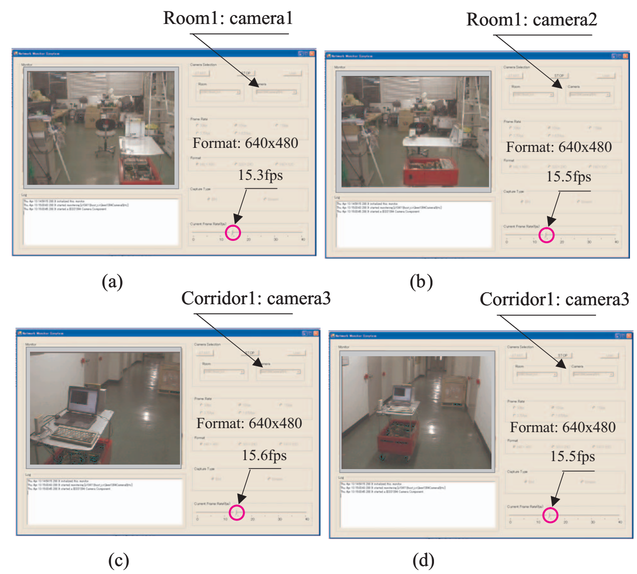


Fig. 11. Experiments of network distributed monitoring system with IEEE-1394 cameras.

the errand robot failed to execute its task three times. The failure was caused because of the measurement error of the robot position.

In order to extend the function of the network distributed monitoring system, we also developed IEEE-1394 cameras RT components. Being different from QuickCam Orbit cameras, IEEE-1394 cameras were widely used in robot field and they have high performance and high resolution. Some performance experiments have been done using the improved network distributed monitoring system to monitor the service robotic system working. IEEE-1394 cameras (Flea, Point Gray Research Inc.) for monitoring the environment were connected to the computer running on Windows XP (2.4GHz, Pentium4), and GUI is run on the other same specification Windows XP (2.4GHz, Pentium4). Two computers were connected in a LAN (1Gbps) and CORBA middleware was used for data communicate between two computers. The average frame rate of video transmission is approximately 15.6fps. Figure 11 illustrates some images of experiments using network distributed monitoring system with IEEE-1394 cameras to monitor the service robot system. By using designed GUI , user can push "Load Button" to select a room name and camera name, a frame rate, a format of image, then push the "Start Button" to start video display. Current frame rate is also displayed in the designed GUI. The user can access any cameras to request video stream without their information about the language of implementation, operating system or network. Figure 11(a), (b) are different video from different camera in the same room. Figure 11(c), (d) are the video from the camera in corridor. When the user push "Stop Button", the display of the video will stop. Experimental results verified that improved network distributed monitoring system with IEEE1394 cameras enhanced flexibility of system and facilitated the integration of different systems.

VI. Conclusion

This paper presented the architecture of the developed RTM-based network distributed monitoring system for human-assisted robot to improve interaction between remote users and local robotic system. Home integration system and network monitoring system using QuickCam Orbit cameras were demonstrated from June 9 to June 19 June at the 2005 World Exposition, Aichi, Japan. In order to enhance the performance of the network distributed monitoring system, we also used IEEE-1394 cameras with high performance and high resolution to develop IEEE-1394 cameras RT components, which makes system applied widely. Some experiments have been done for the improved network distributed monitoring system, and experimental results verified that improved network distributed monitoring system with IEEE1394 cameras enhanced flexibility of system and facilitated the integration of different system. Because Robot Technology Middleware (RTM) was used in the developed system, the system has high scaling and inter-operating ability, facilitating network-distributed software and sharing. making application and system integration easier. For future work, we will develop the other functional robot system components as RT components such as RFID RT object recognition component or RT localization component for localizing mobile robot.

VII. Acknowledgments

The home integration robotic system and network distributed monitoring system were 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.

References

[1] Ando, N., Suehiro, T. et al., Implementation of RT composite components and a component manager, The 22nd Annual conference of the Robotic Society of Japan, IC26, 2004.
[2] kitagaki, k., Suehiro, N., et al., GUI components for system development using RT components, The 22nd Annual conference of the Robotic Society of Japan, IC23, 2004.
[3] Baba, T., Mizukawa, M., Ando, Y., " Function Integration in the Physical Agent System", 7the SICE System Integration Division Anual Conference, 2006.
[4] Jia, S. and Takase, K., "Internet-based robotic system using CORBA as communication architecture", *Journal of Intelligent and Robotic System*, 34(2) (2001), pp. 121-134.
[5] Hada, Y., Jia, S., Takase, K. et al., "Development of Home Robot Integration System Based on Robot Technology Middleware", The 36th International Symposium on Robotics (ISR 2005), Japan.
[6] Inamura, W., Fujii, M., and Murakami, H., " Improvement of Real-time Performance using Composite Component", 7the SICE System Integration Division Anual Conference, 2006.
[7] Object Oriented Concepts, Inc., *http://www.omg.org*.
[8] Java remote method invocation: *http://java.sun.com/products/jdk/rmi/index.html*.
[9] *http://www.is.aist.go.jp/rt/*.