# We are IntechOpen, the world's leading publisher of Open Access books

# Built by scientists, for scientists

# Internet-Based Service Robotic System Using CORBA as Communication Architecture

Songmin Jia, Kunikatsu Takase
*University of Electro-Communications,*
*1-5-1 Chofugaoka, Chofu-City, Tokyo 182-8585, Japan*

## 1. Introduction

Many Internet robotic systems have been developed on the Web over the last few years. Internet telerobotic systems have typically been operated using two ways communication links between a client and a remote robotic system (Barney Dalton et al, 1998), one is using HTTP (HyperText Transfer Protocol) combined with CGI (Common Gateway Interface) and the other way is using Java. The University of Southern California's Mercury Project (Ken Goldberg et al, 2000) used HTTP combined with CGI (Common Gateway Interface) to develop a tele-excavation system that enabled web users to manipulate a remote "real world". CMU project developed Xavier autonomous indoor robot on the Web using CGI (Reid Simmons et al, 1998, Reid Simmons et al, 2000). KhepOnTheWeb (Patrick Saucy et al, 1998, Patrick Saucy et al, 2000) uses CGI, which allows the user to control a Khepera robot in a static environment. Another CGI system is the WebPioneer (WebPioneer, 1998), in which the user drives a Pioneer robot in an unknown room. CGI system causes latency and poor scalability because a new process must be started for each user request. The USA Puma Paint project (Matthew R. Stein et al, 1998, Matthew R. Stein, 2000) is a Web site allowing any user with a Java ™ compatible Web browser to control a PUMA760 robot to paint on an easel with real brushes and paint. Mobile robot on the Web project (Roland Siegwart et al, 1998) uses Java to establish a modular framework for mobile robots on the Web. Java is executable within a Web page. Java programs can be run as applets within the browser and supply an interactive interface instead of a static one. But Java client must know the location of all servers to which it wants to connect and Java allows applets to connect only to the host they were served from. In this paper, we propose using CORBA to implement networking connections between a client and a remote robotic system.

Our goal was to propose the advanced architecture to implement the Internet robotic system that can overcome the shortcomings of the other Internet robotic systems. CORBA uses an Object Request Broker (ORB) as the middleware that establishes client/server relationships between objects. This allows a client to invoke a method on a server across network transparently without needing to know where the application servers are located, or what programming language and operating system are used.

Our secondary goal was to implement the network-based robotic system to assist the aged or disabled in their homes because the elderly population is growing while the number of people to take care of them is declining. The primary task of the system would be to supply food and a cup of water or juice to aged and disabled persons. To



Source: Industrial Robotics: Programming, Simulation and Applicationl, ISBN 3-86611-286-6, pp. 702, ARS/plV, Germany, December 2006, Edited by: Low Kin Huat

accomplish these tasks, the human caregivers can control the telerobotic system to retrieve and manipulate the desired tableware or the other objects, by viewing live image feedback from a remote computer. In our research, we first implement an Internet-based hand-eye robotic system using CORBA as a basic platform. The user can control the remote hand-eye robotic system at a task-level to recognize and manipulate the spoon, the fork, and so on by the intuitive user interface. Then we developed multi-robots to perform service tasks for supporting the aged or disabled. A video/audio conference system based on Robot Technology Middleware (RTM) was also developed to improve the interaction among the users and local robotic system, and enable web-user to monitor the state of robotic system working and the state of the aged or disabled, and to get a better understanding of what is going on in the local environment.
In summary, the paper contributions are:

- Propose using Common Object Request Broker Architecture (CORBA) to implement networking connections between a client and a remote robotic system. This makes the system has the distinct features as following:
  - Clients and servers do not need direct knowledge of each other; CORBA-based system has high scalability to allow clients and server objects, for example written in different languages, run in different operating system, and connected in different network to inter-operate.
  - The CORBA-based Internet robotic system can be implemented by a number of independent components, and these components of the system can be run independently to implement the application and easily be integrated with the other technologies into new comprehensive application systems. This facilitates network-distributed software sharing and improves the cost of writing and maintaining software.
  - The other components of the system can work normally even if there are problems with some of them.
- Develop a CORBA-based Internet hand-eye robotic system and CORBA application servers to allow the remote user to control the robot arm to recognize and manipulate the objects at a task-level.
- Develop multi-functional services robotic system and key technologies to enable the developed system to provide daily service task to support the aged or disabled.
- Experimental results verified that using CORBA to implement networking connections between a client and a remote robotic system is very flexible, effective and robust.

The rest of the paper consists of 7 sections. Section 2 presents Common Object Request Broker (CORBA) in detail. Section 3 describes the structure of the system hardware base. Section 4 details the functions and interfaces of the task-level robot arm control server, live image feedback server, mobile robot control server, and iGPS server. Section 5 introduces the developed network monitoring system based on RT middleware. Section 6 explains intuitive Web-based user interface. Experimental results are given in Section 7. Section 8 concludes the paper.

## 2. Common Object Request Broker Architecture (CORBA)

In our aging society, it would seem particularly important to integrate various kinds of network-distributed software and robotic systems for complicated applications aiding the elderly population. A distributed computing technology that can implement network-distributed software sharing and improve the cost of writing and maintaining software is in high demand.

CORBA combines the benefits of object-orientation and distributed computing and is a middle-ware architecture that provides an object bus that allows components/objects to communicate with each other across networks (Seán Baker, 1997). There are the others distributed computing technologies, like RMI (Remote Method Invocation), DCOM (Distributed Component Object Model), MOM (Messages Oriented Middleware). Sun's Java RMI (Java) is a Java-specific RPC middleware that provides a simple and direct model for distributed computation with Java objects. However, its simplicity is enabled by restricting all communication to Java objects only. DCOM is Microsoft's architecture for distributed component-based communication and is based on COM, which is both a specification and implementation developed by Microsoft Corporation. MOM (mom) provides process-to-process data exchange, enabling the creation of distributed applications. It is analogous to e-mail in the sense that it is asynchronous, requiring the recipients of messages to interpret their meaning and to take appropriate action. MOM is not an industry standard.

In contrast to all of these, CORBA uses an Object Request Broker (ORB) 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.

The main components of CORBA are shown in Figure 1.

- The Object Request Broker (ORB core) is responsible for the delivery of client-requests from one computer to objects possibly on the other computer as well as for the return of the resulting information. By hiding all details of data transmission from the client, it facilitates the communication between clients and objects. To target an object, the client only has to specify the object reference, which is created when the object is created.
- OMG Interface Definition Language (OMG IDL) defines the object interface and specifies the types and operations that the objects support. IDL has features like language independence, distributed service specification, definition of complex data types, and hardware independence. By compiling this interface with an IDL compiler for mapping to client/server source code in a specific programming language, we can get client stub and server skeleton source code for communication between client and server (SJT Condie, 1999).
- Object Adapter (OA) provides a binding between an objects interface and a server's implementation. Object Adapter is responsible for the registration of an object at the ORB and the creation of the object reference. The most commonly used OA is the basic object adapter (BOA), which all CORBA vendors must supply.
- Dynamic Invocation/Skeleton Interface (DII/DSI) allows an application to issue requests for any interface, even to an interface that was unknown at the time when the application was compiled.

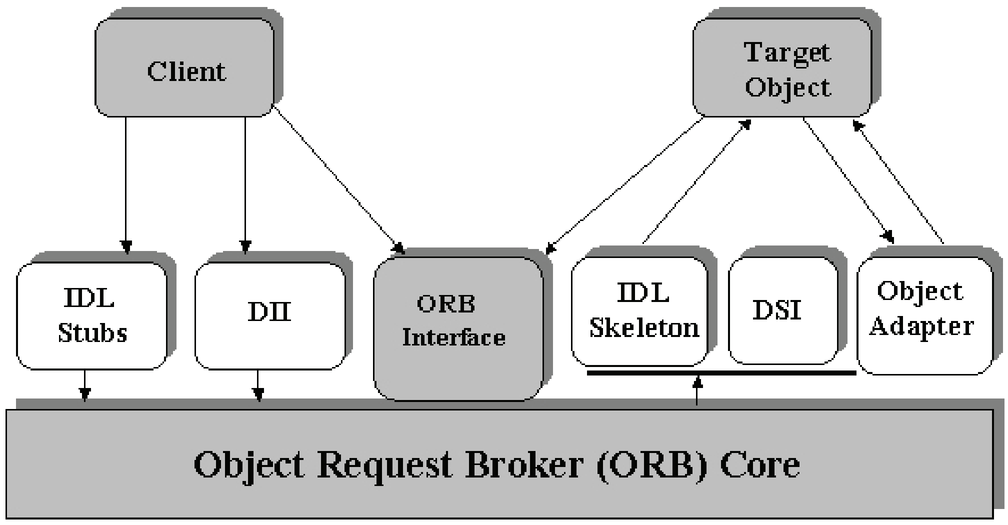



Fig. 1. Common Object Request Broker Architecture.

## 3. System Description

The population-aging problem is increasingly pressing society. According to statistical data from the Ministry of Welfare of Japan, those 65 years or older will comprise 25.2% of the Japanese population in 2015. Meanwhile, the population of the aged worldwide is increasing at approximately a 12% rate. This group requires special care and functional aids due to the progressively reduced degree of autonomy. Because of rising costs and the shortage of nursing home personnel, the current trends for long-term care for the elderly are at-home care or residence at small-scale distributed facilities. Thus, the development of supporting systems to improve care cost and the QoL (Quality of Life) of the elderly could not come at a better time.

Our HARSP (Human-Assistance Robotic System Project) project consists on developing a Human-Assistance Robotic Distributed System in order to provide daily service to aid aged or the elderly people (Jia et al, 2004). The proposed system has the potential to provide elderly persons local services in:

- Intelligent reminding: remind elderly persons about important activities such as taking medical, eating meals, visiting the bathroom, or scheduling medical appointments.
- Data collection and surveillance: many emergency conditions can be avoided with systematic data collection. For instance, robots can support the aged or disabled in using medical peripherals, such as a telephonic stethoscope, vital sign and glucose meters. Robotic systems may be soon able to call caregivers for assistance if they detect that an elderly person has fallen or the other emergency.
- Daily services: many elderly persons with some chronic disease give up independent living because of difficulty in moving their body to take something such as operating objects in a refrigerators. An omnidirectional mobile robot integrating with a skillful robot arm could help the aged or disabled overcome these barriers and supply them necessary daily services.
- Mobility aid: support the aged or disabled for getting up from the bed or a chair, and implement intelligent walking aid.

Multi-robot cooperation is indispensable in order to perform service tasks for supporting the aged or disabled. The mobile platform equipped with a dexterous manipulator is convenient, but it is very difficult to handle the objects (such as operating objects in a refrigerators) because of the difficulty to control the position and orientation of the mobile platform and the manipulator mounted on the mobile platform. In our system, we adopted using a robot arm with five degrees of freedoms cooperating with a mobile robot to implement the service tasks. This method makes it easier to operate the objects such as in a refrigerators. Figure 2 illustrates the hardware configuration of the developed system including robots, cameras, server computer and users' computers.

### 3.1 Robot Arm and Hand

The Mitsubishi Movemaster Super RV-E3J was used in this system. It consists of a manipulator with five degrees of freedoms, a controller, and a teaching box. The maximum speed of the robot arm is about 3500mm/sec; its load weight is about 3.5kgf. It is fixed to a place where there are many objects that need to be manipulated.
In order to implement soft grasp, the robot hand with force sensors has been designed. We also designed the CPU control circuit to measure the grasp force and the servo-driving circuit to drive the fingers. Futaba's S3801 high-torque metal gear servo was used to drive the fingers. Its weight is about 107g, its operating speed is 231°/sec, and its output torque is 14.0 Kg·cm. This servo can act on PWM (Pulse Width Modulation). The basic cycle is 20 ms. We generate input control pulses by a PTC (Programmable Timer Controller) to control the servo. The robot hand works according to the commands coming from the CPU controlled by the server computer (Jia et al, 2001).

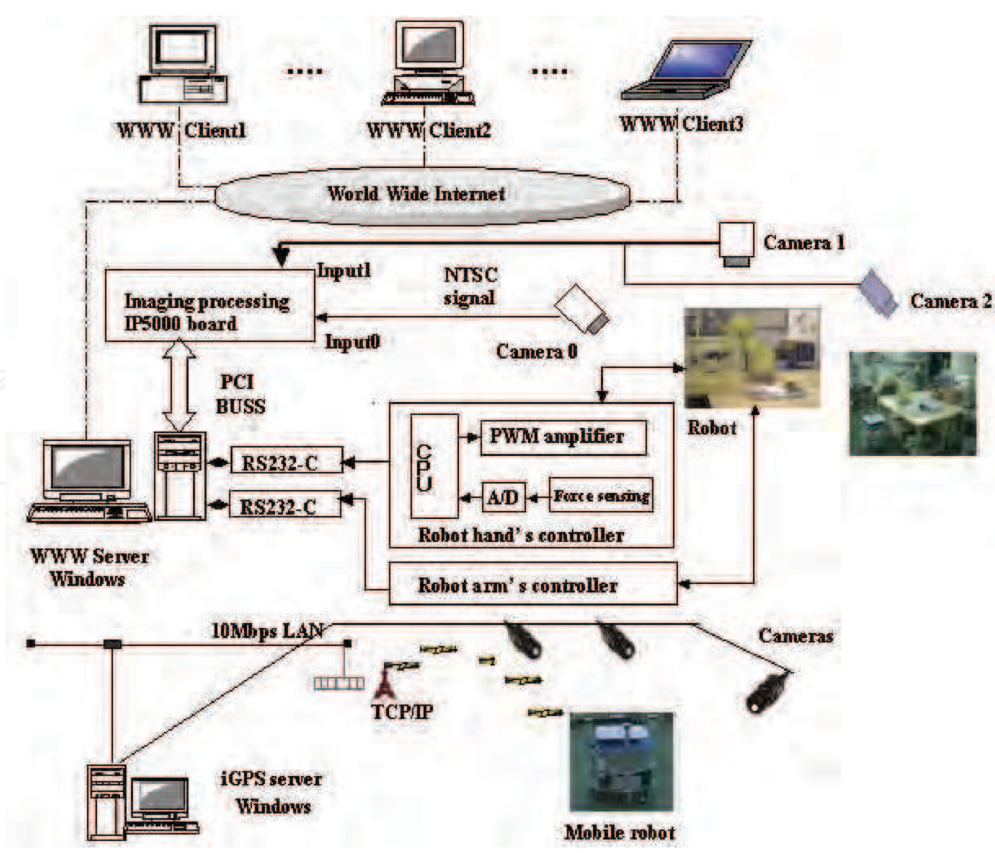



Fig. 2. Hardware base of the developed system.

### 3.2 Vision System

VC-CI MKII cameras are positioned at the place where they can provide user feedback images easy to understand, the other VC-CI MKII camera is used to take an image of the tableware or the other things scattered on the table in this system. A high-speed image processing board, IP5000, was used for processing the obtained images. The resolution of the vision system is 512 x 512 pixels, and it takes 3.6 milliseconds to process each image. For localizing the mobile robot, SONY XC-55 TV cameras with IR ray pass filters are mounted on the ceiling of the environment in which the mobile robot moves . The SHARP GPB-K image processing board processes images and localizes the mobile robot by detecting the IR LED units arranged on the mobile robot (Hada, 2004).

### 3.3 Mobile Robot

An omnidirectional mobile robot was used to deliver pills, juice or other objects to the place where the disabled or aged person lies or sits. It has a diameter of 45cm and stands 66cm tall. It can translate at speeds up to 300 mm and rotate at a speed up to 30 degrees per second. The mobile robot includes a three-wheel mobile base that can move in all directions, a control system to control the motion of the mobile robot and a network interface which supports a wireless TCP/IP Ethernet link (Hada, 2004). To cope with objects vibrating on the tray when the mobile robot is moving, special nonskid materials were used for the tray surface.

### 3.4 Server Computers and User's Computer

The server computer receives client's commands from worldwide network, does various kinds of processing and then sends commands to the devices that can implement the tasks. A high-performance computer is necessary if we want to implement a good-performance system.
For this system, linking to the Internet is the only requirement on user's computer. World Wide Web users can use any computer platform and operating system.
The communications between server and robot arm's controller and the robot hand's controller are through RS232-C links. Users' computers in the world communicate with the server via world wide Internet.

## 4. CORBA Application Servers

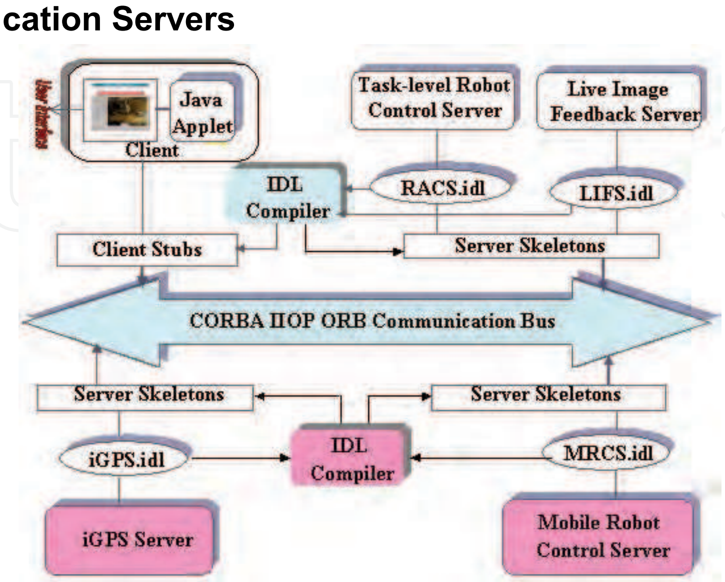
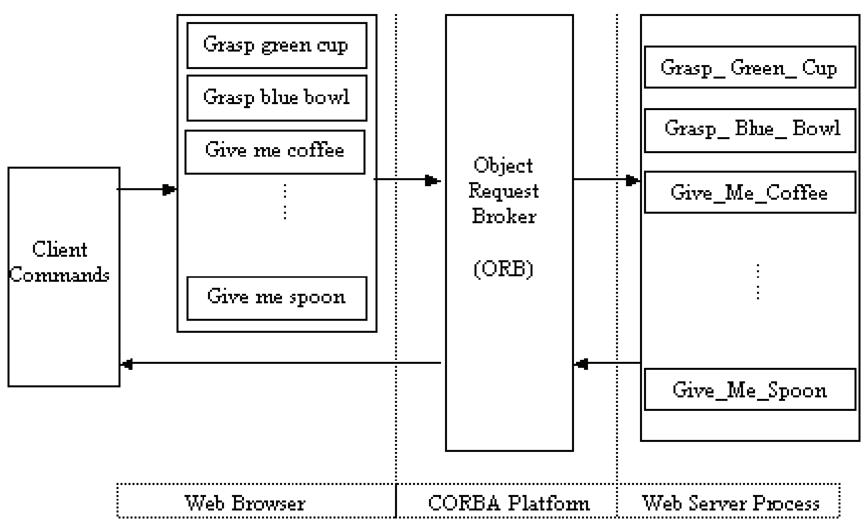



Fig. 3. Network connections of CORBA-based Internet robotic system.

We use CORBA as communication architecture to implement an Internet robotic system. We have implemented task-level robot control server, mobile robot control server and live image feedback server by the optimal programming language C++ because C++ is the most proper language for the core functions of the system, such as imaging processing, robot control. We have also implemented a client by a Java applet because Java applets are capable of directly accessing CORBA application servers via the Internet Inter-Object Request Broker (IIOP) and are easy to run within a Web browser that can be used by non-specialists from any Internet site. The network connections of CORBA-based Internet service robotic system are shown in Figure 3.

### 4.1 Task-Level Robot Control Server

Task-level robot control server allows the caregivers to control this remote robotic system at a task-level (Jia et al, 2002). The user can invoke the methods on this application server across the Internet to recognize and manipulate the tableware or the other object what they want. The assignments of the server is as follows (Figure 4):




| Interface Name | Functions |
|---|---|
| Handle_ Cup | Provide service for manipulation of cups |
| Handle_ Bowl | Provide service for manipulation of bowls |
| Handle_Can | Provide service for manipulation of cans |
| ⋮ | |
| Handle_ Spoon | Provide service for manipulation of spoons |

Fig. 4. Robot control server and its interface.

- Receive commands from the remote World Wide Web users.
- ORB intercepts the "call" and is responsible for finding an object that can implement the request, passes it the commands.
- Find the location and orientation of the tableware or the other objects scattered on the table by the vision system.
- Generate the task implementation plan and send the commands to the devices.
- ORB returns the feedback results to the remote clients.

For one method on the task-level robot control server, it consists of the following parts (Figure 5).

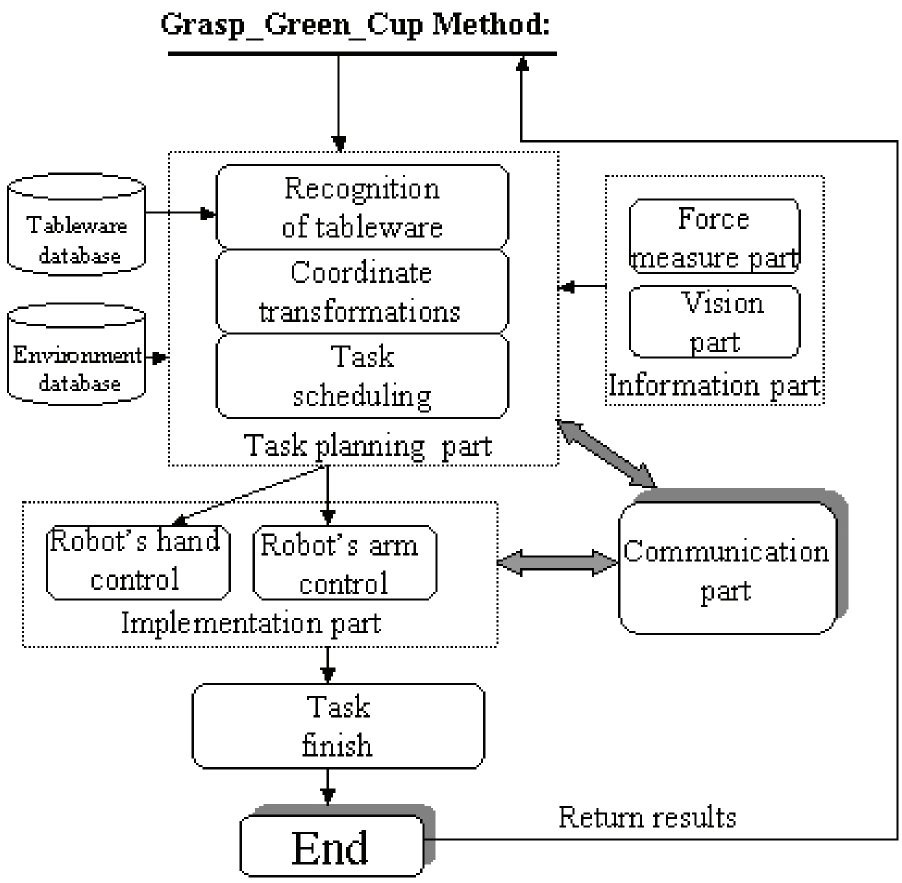


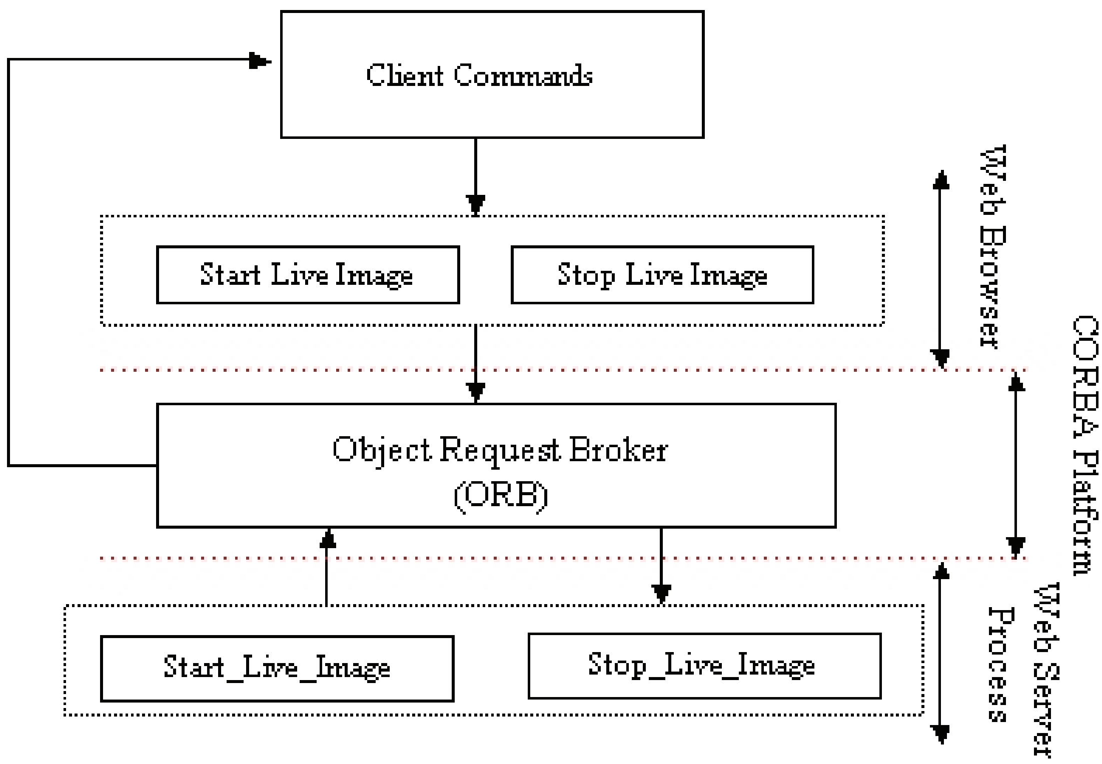

Fig. 5. Method of robot control server.

- The information part consists of a vision part and a force measure part. It is the source of information for the system.
- The task planning part receives the information from the information part, recognizes the location and orientation of the tableware or the other objects scattered on the table, transforms these coordinates to the manipulator's coordinate system, and generates task plan to achieve the goal.
- The implementation part executes motion scheduling generated by the task planning part, and it implements the task according to the commands coming from the server computer.
- The communication part is responsible for the communication between the server computer, the robot's arm and the robot hand's controller via RS232-C links.

For real-time recognition of circular tableware, the quick correlation calculation was used. Correlation calculation is a method that evaluates the matching between the template image registered beforehand and the input image obtained from the camera.
For long tableware or long objects such as spoons and forks, we assume that they consist of two different colour parts. Using this technique, long tableware or long objects recognition can be replaced by the simple mark recognition because there is one-to-one correspondence between tableware or objects and different combination of colour marks. We can create the tableware 's or the other objects' geometric model and entry them beforehand. By doing this we can recognize the tableware or the other objects easily and quickly.

### 4.2 Live Image Feedback Server

Live image feedback server provides live image feedback from the camera for the remote user and continuously sends live images of the working robot to the client according to the user's requests (Figure 6). It works as:




| Interface Name | Functions |
|---|---|
| Contol_Live_Feednback_Image | Provide live feedback image service |

Fig. 6. Live image feedback server and its interface.

- Receive image control commands from the remote users.
- ORB intercepts the commands and transfers the requests to live image feedback server.
- Get the new image by camera0 and change the image into a BMP format by IP5000 image processing board.
- Compress this image into JPEG format.
- Return the new image with the last information to the client by ORB.

### 4.3 Mobile Robot Control Server and iGPS Server

In order to realize quick response to a request from a client, we implemented a distributed iGPS server run on another computer (iGPS server computer, connected in 100BASE-TX LAN, running Windows) to share the calculation of the position and orientation of the mobile robot.
The mobile robot control server allows the remote user to specify the best route for the mobile robot and control the mobile robot when mobile robot execute a service task in the real environment. It works as follows:

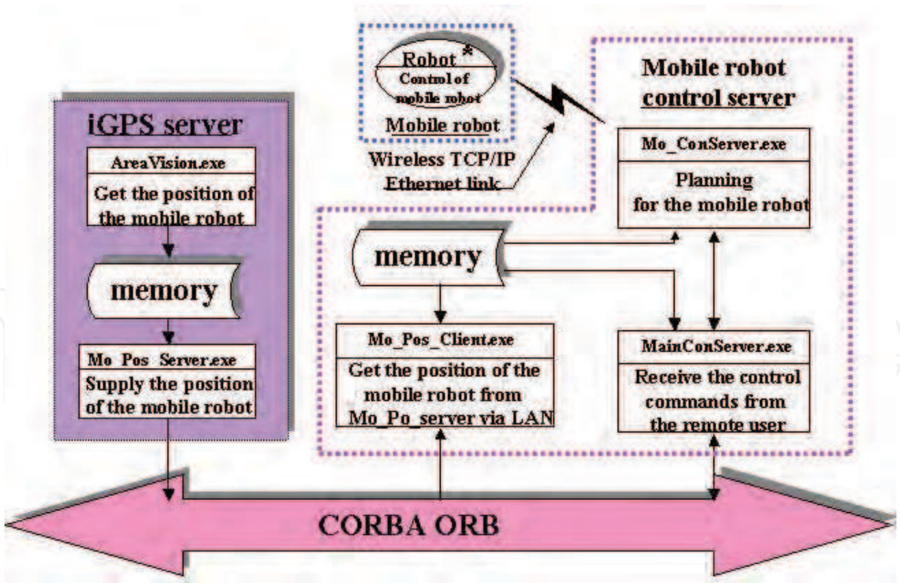


Fig. 7. Mobile robot server, iGPS server and their configuration.

- The mobile robot control server receives control commands from the remote World Wide Web users.
- ORB intercepts commands and transfers the requests to the mobile robot control server.
- The mobile robot control server plans and derives a collision-free path that is the shortest distance between the START and GOAL specified by the user.
- The mobile robot control server programs the omnidirectional mobile robot to move across wireless TCP/IP Ethernet link.
- ORB returns the feedback results to the remote clients.

Figure 7 illustrates Mobile robot server, iGPS server and their configuration. By viewing live feedback images the user can determine whether the mobile robot has arrived at a place where it can easily receive an object from the robot arm or pass the object to the disabled person. If there is displacement, the user can adjust the position of the mobile robot by fine adjustment commands using interface. Similarly, the remote user can also rotate the mobile robot to a certain angle.

## 5. Network Monitoring System Based on RT Middleware

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 in network distributed human-assistance robotic system. Network monitoring system using QuickCam Orbit cameras was implemented based on RTM. RT Middleware framework was developed by AIST (Agency of Industrial Science and Technology, Japan) to promote application of Robot Technology (RT) in various fields. RTM is based on CORBA (ominiORB), so the components of the system can be implemented by different programming languages, run in different operating system, or connected in different networks to inter-operate. Each application is made up of components; integration is supported by allowing other applications to communicate directly with these components. This facilitates network-distributed software sharing and improves the cost of writing and maintaining software.
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). We develop robotic functional elements as "RT software component", which makes application and system integration easier.

The QuickCam Orbit cameras were used in our system with high-quality videos at true CCD 640x480 resolution and automatic face tracking and mechanical Pan, Tilt 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. Two QuickCam Orbit cameras were set up in the environment, and they can overlook the area the service robots moves in by adjusting the mechanical Pan and Tilt of the cameras. The structure of the developed network monitoring system was shown in Figure 8.

In addition, we developed a graphic user interface (GUI) for the video stream system that provides a remote video stream (format 320x288 was selected because image data transition's problem), 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. 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.

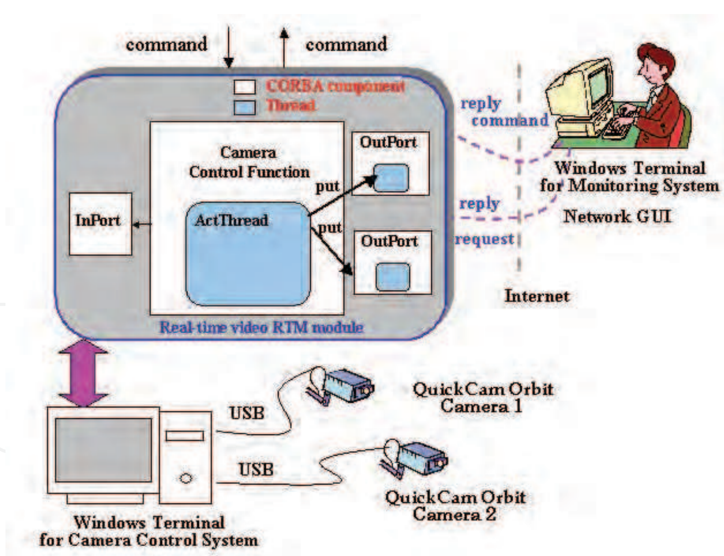



Fig. 8. Structure of network monitoring system based RTM.

## 6. Intuitive Web-Based User Interface

The challenge of designing a user interface is to provide enough information to let the user handle this system easily while minimizing transmitted data and maintaining a simple layout. Besides that, all kinds of complex calculations should be hidden from the user in order to support a wide range of users. The Web-based user interface designed is shown in Figure 9.

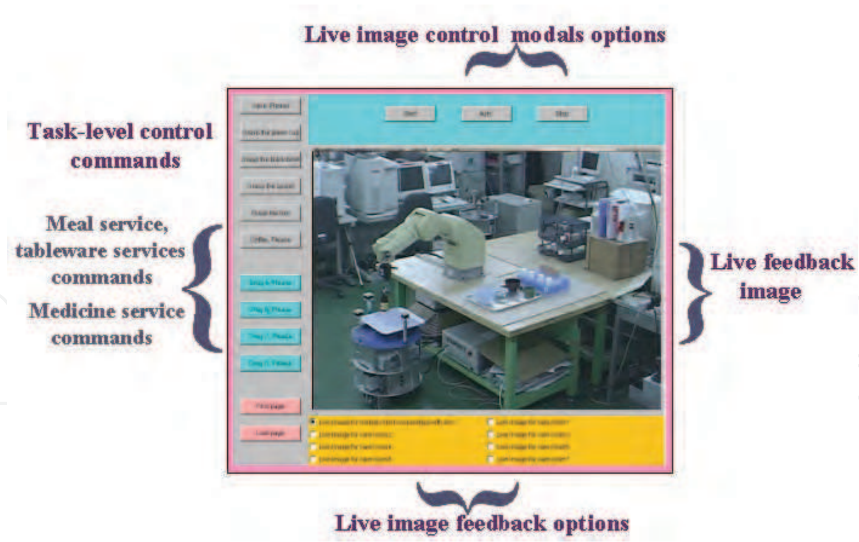


Fig. 9. Web-based user interface of the developed system.

User interface designed consists of live image feedback part, options for different kinds of live feedback images part, and task-level robot arm control commands part. The task-level robot arm control commands allow the user to submit a task-level request to the system. It consists of meal service commands, drug service commands, and a common service command such as providing a book. These commands include ``Grasp the green cup," "Grasp the spoon," Juice, please" etc. Once one task-level button is pushed, the other task-level button will be invalid until this task is finished for safety. When the user push the button "grasp the spoon", this command will be sent to server and the necessary result message will be feedback to the client by popping one message text box on the Web-based user interface. Many options for different kinds of live feedback images have been provided, including images of the mobile robot cooperating with the manipulator, images of the state of the rooms of the disabled or aged. In addition, "auto" and "step" control modals of the live image feedback are provided to allow the user to see the continuous live feedback images or the "step" image feedback that refreshes the image once button is pushed.
When the remote user links to this homepage, a geometric 2D map models the environment of the robot system's working domain. Its geometric primitives can also be adjusted to add new obstacles if the environment of the robot system has been changed. The positions where the mobile robot can easily cooperate with the robot arm and pass objects to the disabled or aged are displayed as marks. The "Remove last point" button's function is to remove the last command that the user specified if the user wants to change the route. The "Remove all points" button allows the user to clear all the commands that the user specified and renew the route. In order to know the real position of the mobile robot in the indoor environment, the real trajectory of the mobile robot is also shown on the interface homepage. The "Up", "Down", "Left", and "Right" buttons allow the user to adjust the position of the mobile robot to facilitate cooperation with the robot arm. The "Rotate" button can also be used to control the mobile robot to rotate.

## 7. Experimental Results

- Performance Evaluation of CORBA

CORBA uses an Object Request Broker (ORB) as the middleware that establishes client/server relationships between objects. The client can invoke a method on a server object across a network transparently without knowing where the application servers are located, or what programming language and operating system are used. This lets system overcome the shortcomings of the other Internet robotic systems. In addition, the components of the CORBA-based system can be implemented and run independently to implement the application, and also be integrated easily into new systems.

The experiments of evaluating the performance CORBA have been done. The client located at PC1 Pentium III, 600 MHz, and NT 4.0) sends a remote method call with 64 long words (4 bytes/word) to CORBA server (located at PC2, Pentium III, 700 MHz, NT 4.0) in a LAN (100BASE-TX), and the CORBA server returns the identical data to the client. This cycle is repeated 1000 times, which is called one set.Twenty sets have been done and the best performance is chosen to be the representativeresults from 20 sets, and its average of the round-trip time is about 0.29ms.The same experiment has also been done with ten simultaneous users, and there is no significant time difference between the result of only single user and that of multi-user.

We also have done the experiments such as: the task-level robot control server is working and the live image feedback server was down. In such a case, the user can still access the robot control server normally. Similarly, the user can still access the live image feedback server normallyeven though the other robot control server was down.

- Remote Accesses to the Developed Hand-Eye Robotic System

We useCORBA as a basic platform and connect our hand-eye robotic system to the Internet. The Web-based user interface was designed and it consists of live image feedback part, robot commands part, and message text box. This system has been accessed by the operators from Tama City, Saitama City, Kyoto University, Japan and Chicago, America etc. These users' computershave different specifications and network connections, and work on the different operating systems. Users operate this remote robotic system toretrieve and manipulate the table ware that they want and they can also monitor the state of the robot executing the task by viewingthe live image feedback coming from the robot site.

Fig. 10. Remote access to the hand-eye system; The remote user uses the telerobot system to manipulate the juice, fork, green cup and vitamin drink and put them to the tray mounted on the mobile platform.

When the remote user pushes the command "Juice, Please", the system can automatically find the juice and execute this task robustly. Figure 10 shows some on-line images that the remote user is accessing the developed hand-eye system to manipulate the fork; green cup and vitamin drink and put them to the tray mounted on the mobile platform. When the users pushing the "start" button, the new image with the latest information will be shown.

- Remote access to the developed multi-robots system to perform a service task.

Figure 11 illustrates some on-line images of the user operating the multi-robots to perform a service task. We developed a hardware base including a robot arm and an omnidirectional mobile robot and CORBA application servers including a task-level robot arm control server, live image feedback server, mobile robot control server and iGPS server, all of which can be distributed on the internet and executed in parallel.

By remotely controlling a mobile robot to cooperate with a robot arm using user interface designed, the developed system can realize some basic services to support the aged and disabled. When the operator links to the homepage, the operator first specifies the best route for the mobile robot and directs it to move to the place where it can cooperate with the robot arm by using the "specify mobile robot route" user interface. By live image feedback or video stream, the user can observe whether the mobile robot has arrived at a place where it can receive objects passed to it by the robot arm. If there is displacement, the user can adjust the position and rotation of the mobile robot. Then, the user operates the robot arm to manipulate the objects requested by the disabled or aged such as food, medicine or tableware. Lastly, the user can direct the mobile robot to move to the place where the disabled or aged person is. Of course, if the remote user can use the task-level control command, the developed system can perform a service task autonomously.

Fig. 11. On-line images remote user interacting with mobile robot to realize a local service task.

In order to enable a remote user to get a better understanding of the local environment, 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 the robotic system working, the state of the age and disabled, and to enable the remote user to know what is going on in local environment.

## 8. Conclusion

We used CORBA as communication architecture to implement networking connections between a client and the developed hand-eye robotic system. CORBA uses an Object Request Broker (ORB) as the middleware that establishes client/server relationships between objects. The client can invoke a method on a server object across a network transparently without knowing where the application servers are located, or what programming language and operating system are used. This lets system overcome the shortcomings of the other Internet robotic systems. In addition, the components of the CORBA-based system can be implemented and run independently to implement the application, and also be integrated easily into new systems. This facilitates network-distributed software sharing and improves the cost of writing and maintaining software. The other components of the system can work normally even if there are some problems with some of them. So, using CORBA to implement Internet robotic system is very effective, flexible and robust.

Using CORBA as communication architecture, we also developed a hardware base including a robot arm, omnidirectional mobile robot system and CORBA application servers including a robot arm control server, live feedback image server, mobile robot control server and iGPS server. By user controlling a mobile robot to cooperate with a robot arm using Web-based user interface, the developed system can provide some basic services to support the aged or disabled.

Network monitoring system based on RTM using QuickCam Orbit cameras was developed to transmit media streams in real time to improve interaction in network distributed human-assistance robotic system. It can enable remote users to get a better understanding of the state of the robots working and the state of the aged or disabled.

Some experiments that the remote user accesses the developed system to provide a service task to support the aged and the elderly people have been done. And experimental results verified the effectiveness of the developed system. For future work, improvements of the navigation of mobile robot and recognition of the obstacles using RFID technology and stereo camera are a main topic. In addition, we will develop the other additional function of the system as RT component in order to improve the flexibility of the system.

## 9. References


Barney Dalton and Ken Taylor (1998). A Framework for Internet Robotics, *Proceedings of 1998 IEEE/RSJ, Conference on Intelligent Robots and Systems; Workshop Web Robots*, pp. 15-23Victoria, B.C. Canada, October.

Hada, Y., Takase, K, Ohagaki, K. et al. (2004), Delivery Service Robot Using Distributed Acquisition, Actuators and Intelligence, *Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 2997-3002.

Java remote method invocation: *http://java.sun.com/products/jdk/\\rmi/index.html*.

Ken Goldberg, Steven Gentner, Carl Sutter, et al. (2000), The Mercury Project: A Feasibility Study for Internet Robotics, *IEEE Robotics and Automation Magazine,* 7(1), pp. 35-40, ISSN.

Maja Matijašević, Kimon P. Valavanis, Denis Gracanin et al. (1999), Application of a Multi-User Distributed Virtual Environment Framework to Mobile Robot Teleoperation over the Internet, *Machine Intelligence and Robotic Control*, 1(1), 11-26.

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

Matthew Stein (1998), Painting on the Web, The Puma Paint Project, *Proceedings of 1998 IEEE/RSJ*, pp.37-43, *Canada*, October 12-17.

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

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

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

Patrick Saucy, Francesco Mondada (2000), Open Access to a Mobile Robot on the Internet, *IEEE Robotics and Automation Magazine*, 7(1), 41-47.

Patric Saucy (1998), KhepOnTheWeb: One Year of Access to a Mobile Robot on the Internet, *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems; Workshop on Web Robots*, pp. 23-31, Victoria, B.C. Canada, October.

Reid Simmons, Joaquin L. Fernandez, Richard Goodwin et al. (2000), Lessons Learned from Xavier, *IEEE Robotics and Automation Magazine*, 7(2), pp. 33-39, ISSN.

Reid Simmons: Xavier (1998), An Autonomous Mobile Robot on the Web, *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems; Workshop on Web Robots,* pp. 43-49, Victoria, B.C. Canada.

Roland Siegwart, Cedric Wannaz, Patrick Garcia et al. (1998), Guiding Mobile Robots through the Web, Proceedings *of 1998 IEEE/RSJ Conference on Intelligent Robots and Systems,* Workshop on Web Robots, Victoria, B.C. Canada, October 12-17 1998, pp. 1-7.

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

Seán Baker (1997), CORBA Distributed Objects Using Orbix, ACM Press, UK.

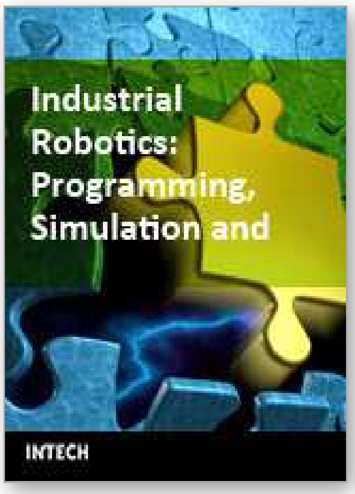
SJT Condie (1999), Distributed computing, tomorrow's panacea-an introduction to current technology, *BT Technol. J*, 17(2), pp.13-23.

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

Songmin Jia, Yoshiro Hada, Gang Ye and Kunikatsu Takase (2002), Network Distributed Telecare Robotic Systems Using CORBA as Communication Architecture, *Proceeding of 2002 IEEE International Conference on Robotics and Automation*, pp. 2002-2007, USA

Songmin Jia, Weiguo Lin, Kaizhong Wang, Takafumi Abe, Erzhe Shang, and Kunikatsu Takase (2004), Improvements in Developed Human-Assisting Robotic System, *Proceedings. Of International Conference on Intelligent Mechatronics and Automation*, Invited paper, pp. 511-516.

WebPioneer Website, ActivMedia, Inc. (1998). http://webpion.mobilerobots.com.




**Industrial Robotics: Programming, Simulation and Applications**
Edited by Low Kin Huat

ISBN 3-86611-286-6
Hard cover, 702 pages
**Publisher** Pro Literatur Verlag, Germany / ARS, Austria
**Published online** 01, December, 2006
**Published in print edition** December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.