Journal of
INDUSTRIAL TECHNOLOGY

Volume 20, Number 3 - June 2004 through August 2004

Controlling Robot Through Internet Using Java

By Mr. Ravindra Thamma, Dr. Luke H. Huang, Dr. Shi-Jer Lou and Dr. C. Ray Diez

Peer-Refereed Article

KEYWORD SEARCH

CIM
Distance Learning
Internet
Manufacturing
Robotics
Teaching Methods

Controlling Robot Through Internet Using Java

By Mr. Ravindra Thamma, Dr. Luke H. Huang, Dr. Shi-Jer Lou and Dr. C. Ray Diez

Mr. Ravindra Thamma. MSIT, is an Assistant professor in the Department of Technology at University of North Dakota. He conducts research on robotic, intelligent controls system, computer interface and ultrasonic. He teaches electro mechanical fundamentals, digital integrated circuits, PLCs, micro controllers, and robotics.

Dr. Luke H. Huang, CIT, is an assistant professor and the coordinator of manufacturing in the Department of Technology at the University of North Dakota. Dr. Huang received his Ph.D. in Industrial Education and Technology from Iowa State University. He loves to work for on improvements in automation and manufacturing systems.

Dr. Shi-Jer Lou is an associate professor in the institute of vocational and technological education at National Pingtung University of Science and Technology in Taiwan. His teaching and research interests are multimedia, computer-aided instruction, knowledge management, and intelligent manufacturing system. He currently serves as the main researcher of the NSC project in Taiwan.

Dr. C. Ray Diez, DIT, is an Assistant Professor and Chair, coordinator of the Occupational Safety and Environmental Health program and Director of graduate programs in the Department of Technology, University of North Dakota.

Background and Need

An issue with people who want to pursue an education is how to enroll in courses while working full-time. While working everyday diminishes opportunities for on campus study, distance education becomes a major option and has been accepted widely. Though distance education works for many fields of study, it is difficult for those fields that require laboratory activities, such as manufacturing. To be involved in laboratory activities, one usually has to travel to the lab. Investment of both time and expense on travel is a big concern for potential students in the field of manufacturing.

The advancement of Internet technology tools developed in 1990s made it possible to access a lab at distance. The Internet technology tools allowed development of the idea of an Online Laboratory (OL). Obviously, an OL is beneficial to distance education because it provides students with instant access to a remote lab without investment in both travel time and expense. The benefit of an OL is that enrollment of distance education in manufacturing courses and degree programs will be expected to increase.

An online laboratory would be maximized by remote control technology through Internet. A major step in the online control came with the experiments using the Internet done by Kuchlin (1997) with Highrobot Telerobotics. The experiments resulted in the development of a software model based on an object-oriented concept that could interact remotely with robot programs. Kuchlin also explored the extensive capabilities of Java with Transmission Control Protocol and Internet Protocol (TCP/IP). In the experiments, a Java program could run on arbitrary remote computer platforms interacting with a robotic control.

In a study conducted to access a remote manufacturing system through internet, Calkin and Parkin (1998) developed simulation tools for remotely-controlled robots. The simulation tools for the robotic hardware were developed using Java and VRML 97 to create a desktop virtual-reality environment. A VRML browser was developed by the researchers to enable a client computer to request transmission of web pages from the host computer. However, a drawback was that the client computer had to have the VRML browser and V-collide's libraries installed in order to communicate with the host computer. Many users were inconvenienced because of the installations. In addition, the VRML browser and V-collide's library required a computer with more complicated configurations and a larger amount of memory.

Rodrigues de Queiroz and Bergerman (1998) attempted remote control operation at the Automation Institute in Brazil. Based on their research, the Automation Institute proposed the first Brazilian Robotics and Computer Vision Virtual Laboratory (RCVVL). In the RCVVL, communication between the user and the remote laboratory was accomplished through a graphical interface which was developed with Java and Hypertext Markup Language (HTML). The use of such programs made the RCVVL accessible worldwide via Internet. Users could access the remote laboratory without having to download executable source codes. The RCVVL utilized two cameras, one on top of the robot and the other on the ceiling of the laboratory. Current images were updated at a rate of three frames per second. This research demonstrated the ability to visibly control a robot through Internet from a remote place. Nevertheless, the researchers did not design the RCVVL to run a program from the client site. In addition, the researchers did not make source codes available and the updating rate of the image was too slow. Thus, there was a clear need to develop a workable web-based control system to promote distance education.

The objective of this study was to develop a prototype Web-based Controlling System (WCS) that could serve as a remote control tool for an unmanned manufacturing cell in an educational laboratory. This system is designed to serve as an OL for distance education. The WCS was comprised of hardware components interfaced with Internet tools to control and program the unmanned cell through Internet from a remote site. Students were able to interact via Internet with the un-manned cell to carry out technological activities. The WCS, which was programmable through the client's computer, was compatible with any operating system (OS) and provides simultaneous text and visual feedback at a visual frame rate of 1 frame/sec.

Hardware Setup

The proposed WCS consisted of a computer, a relay board, PLC devices, robots, actuators such as a conveyor belt, and cameras. The working configu-ration is illustrated in Figure 1. The computer, which connected all other components together, was the control center as well as the web server that enables access from client computers through internet for remote controlling.

The computer was equipped with server software called VQ Server. The VQ Server applied in this study was version 1.9.55. It was specialized to enable creation of host web pages in HTML format for access from client's computers. The computer was also equipped with Web Cam 123 to receive images from a digital camera and an author-written program for overall control, including the relay board.

The relay board was connected to the computer via a parallel port. There were 12 relays on the board, which were controlled by the computer through a control program, to manipulate motions of a robot and a PLC program. In turn, the PLC program activated or stopped the conveyor. The robot worked at the conveyor, loading and unloading. The process was monitored with a digital camera, which sent image pixels to the computer to make the control process visible at the remote site.

This setup enabled a client computer to receive, via internet, a web page (Figure 2) that includes all control tools such as buttons to control the motions of the robot and the conveyor, instant images from the digital camera to monitor the system, a text box to show the status of the system, and a window to browse and send a program to run the system.

Development of Controlling Programs

Several computer programs were developed to accommodate the WCS. Interactions of these programs are illustrated in Figure 3.

1. Programs for the Relay Board

The manufacturer of the relay board, DIY Electronics, provided a program called Relay.exe in Borland Turbo C 3.0 language. Relay.exe had a sub routine to identify parallel port one

Figure 1. Setup of the Web-based Controlling System (WCS)

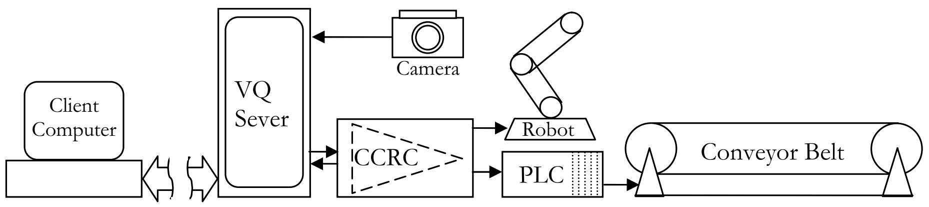
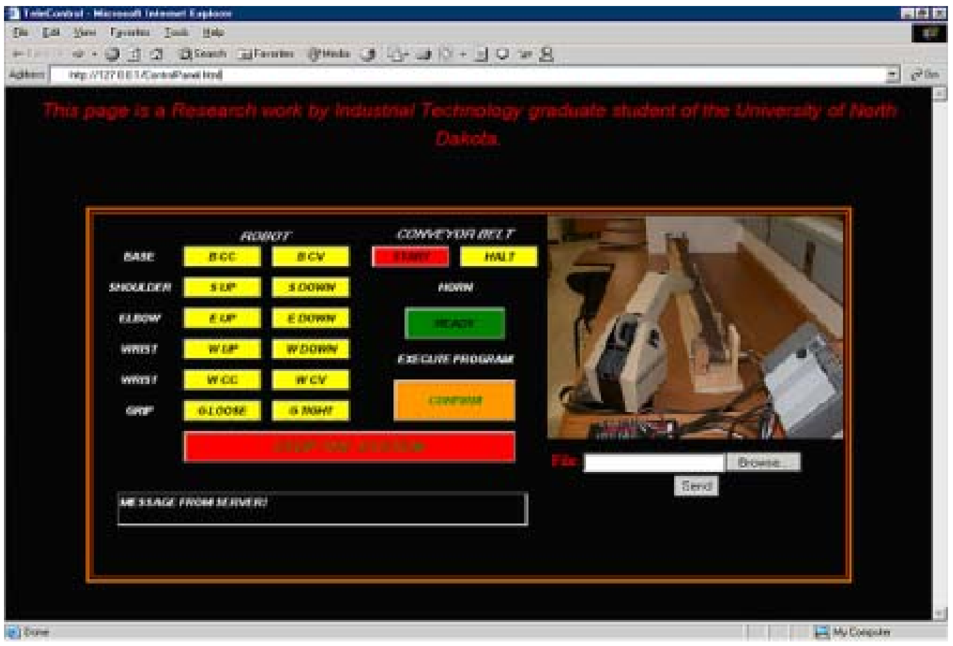


Figure 2. A screen view of the webpage



This page is a Research work by Industrial Technology graduate student of the University of North Dakota.

MESSAGE FROM SERVER!

(LPT1) and execute a 5-volt signal to the requested relay. In this research, the program was modified to be able to execute the requested relay on a specified parallel port. The manufacturer of the relay board also provided another executable file, called Delay.exe, which enabled a specified interval of time to pass before the execution of the next command. Batch files were written to run these executable programs. When the batch files were executed, they activated the output signals to the parallel port on the server. These signals were transmitted through the parallel port to the relay board to make contacts on the respective relays.

Two batch files were developed to provide an emergency stop for the entire system. One batch file executed the program that was developed and transferred to the server side by the end user and the second batch file broke the contacts in all relays.

2. Client/Server Programs

The client/server program was developed with Java (version JDK 1.3 from Sun Microsystems) using TCP and Socket programming with a text feedback to the client's computer. Since client programs were written in Java, they could be accessed from any WWW browser and ran locally on a user's computer. The client program consisted of buttons, a textbox, and network sockets, which were designed to send and receive data from the server. A total of 17 buttons and one textbox were developed in the client applet (Figure 2). These 17 buttons were configured as:

- 12 buttons to control the robot,
- 2 buttons to control the conveyor belt,
- 1 button to activate a horn on the server side,
- 1 button to deactivate the entire system for emergency purposes, and
- 1 button to initiate a program on the server side.

The textbox displayed messages from the server side as well as messages naming the next action to take place at the server side. The network socket was used to transmit data from the client to the server. When a button was depressed, a data string was sent through the socket to the server.

The server program, upon receiving the data strings from the client, converted the string into integer values and passed the integer values to next step. In this step the switch structure was reviewed and executed based on the appropriate batch file. After executing the batch file, the server returned the integer value to the client indicating the successful execution of the command. Figure 2 depicts the client side screen.

3. Robot Interface

A robot, manufactured by Microbot, Inc., was interfaced with the Relay Board. The wired control panel of the robot had push buttons to move the robotic arm. A modification was made to the wired control panel by interfacing it with the relays, which duplicated the wired control of the robot. This provided a second set of controls that was the same as the first set of momentary switch controls provided by the manufacturer. Whenever a relay made a contact, the circuit was completed and the robotic arm moved as commended. The appropriate relays would make contacts on execution of the appropriate batch files.

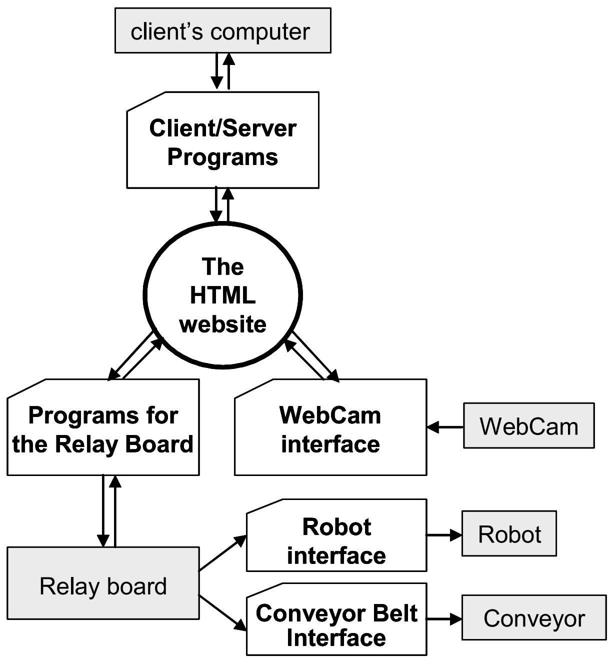
4. Conveyor Belt Interface

A pre-designed conveyor belt was employed to interface with the server through the relay board. The conveyor belt was controlled through a PLC Controller (Allen Bradley SLC-500/3). Ladder logic programs contain latch and unlatch bits. Whenever a signal was given to an input bit the output circuit was completed and stays in a latched condition. With the bit in the latched condition the circuit remains closed and the conveyor belt begins rotating. Giving the second signal to the input bit unlatched the circuit, which in turn, stopped the conveyor belt. Activating relays transmitted signals to the input bit of the PLC.

5. The Website

A web site was developed with HTML to transfer files between the client computer and the server computer. The website can be accessed through any WWW browser so that the robot and the conveyor belt could be programmed and controlled via the website.

Figure 3. Interactions of system programs



6. The program for visual feedback
On-site visual feedback was enabled through the use of a Web Camera via internet. The software, WebCam 123 provided by the WebCam Corporation, was employed to receive the on-site pixels. These pixels were integrated into the Website HTML file and sent to the client computer to enable the visual aid.

WEBCIMS Software Description

The Java Web CIMS software had two main parts: the client/server program and the Relay program. The C language was used to write relay programs to activate the relays through the parallel ports. Java language was used to write client/server programs.

The client /server involved TCP/IP communication using TCP/IP stream sockets. It was a reliable, two-way data connection between two applications. TCP/IP stream sockets delivered sequenced and unduplicated data (i.e., the data bytes were received in the same order that they were sent). Java programming platforms provided classes that create stream sockets.

In TCP/IP communication, the web server delivered the Graphical User Interface (GUI) program to the client and the client GUI communicated directly with the server program through TCP/IP streaming ports. A direct communication link was created between each client and the server. When using this programming method, Java was the rational programming platform for the client GUI. Java applets could be delivered via web pages and executed in most popular web browsers, including Netscape Communicator and Internet Explorer.

Upon request of data from the client computer, the server sent out the appropriate information to the client. The web page on the client computer displayed the Java buttons. The server program sat and waited for user commands to be selected on the GUI (i.e., clicking the button or buttons). When a button was selected, it opened port 2346 and sent commands in the form of a data stream to the server. Before data were sent, a numeric variable was placed in a character string and transferred as a data stream. Each button had a different numeric variable for communication. This port was then closed after all data were transferred.

During the process of data transfer, the text field displayed a message identifying what action would take place at the server side after the process was complete. The server received the data stream by opening port 2346, after which the character string underwent a change. The character string was then converted into an integer. The converted integer was passed to the switch control structure. The switch control structure evaluated the value of integer and triggered the appropriate case depending on the value of the integer. As the case was triggered, it executed a batch file.

When the batch files were executed, the appropriate output signals were sent to the parallel port on the server. These signals were transmitted through the parallel port to the relay board to make contacts on the relay. As the relay made a contact, the robot controller received the signal and instructed the robot to move as specified by the signals. The end user could observe the on-site robot's motion through a web camera.

After successfully executing the C program, the server sent the data stream to the client program. This integer value was then placed in a character string and returned as a data stream. The client program, upon receiving the string, displayed the returned data in a text field as a conformation number. Figure 3 depicts the operation of the WCS.

Performance of the System

The system as described in the Hardware Setup section and Figure 1 was built and configured with a Dell computer (Pentium 333, 64 Mb RAM) as the server. When the system was turned on, any other computers, either on campus or off campus, could access the server and execute commands or programs. The working process could be watched on the small window on the webpage as shown in Figure 2. The robot could be controlled either with manually pressing buttons on the webpage or by sending a program. The conveyer belt could also be turned on and off by pressing buttons on the webpage. The whole system could be shut off by pressing the "Stop the System" button on the webpage when needed. With all these evidences, it can be concluded that the system works and functions as it was designed.

Conclusion

A WCS was developed in this research to remote control laboratory activities through Internet. Special features of the WCS allow end users to use a general-purpose personal computer and any World Wide Web browser to carry out the control activities remotely through Internet. The system also provided feedback to the end users in terms of visual and text format.

Since the configuration of the system as shown in Figure 1 was able to serve as a WCS, similar configuration can be applied for other lab setups. As designed, the client/server program written in Java using TCP and Socket programming made the server accessible through Internet by using a universal browser.

However, due to the unexpected traffic in the Internet, delays of the communication between the server and the end user may happen. This causes delays on executing commands and programs and transmitting on-site images.

Recommendations For Future Work

The following are a few recommendations that can be implemented to enhance this online control study in the future:

1. Use high-speed network to reduce the delay time between the transitions of streams from client to server.
2. Solid-state devices can be used to replace the electromechanical services.
3. Multiple Cameras can be used to monitor the process from multiple

angles to prevent blind spots.

4. Software that transfers a source code from client computer to the server, compiles it at the server side, and then executes the program safely needs to be developed.
5. A proof reading method can be adapted to execute the transferred program safely within the Robot's operating limits. It can be arranged by using Artificial Intelligence and neural networks. This research can be further enhanced by, controlling other devices, not merely Robots and PLC.

References

Calkin D. W., & Parkin R. M. (1998). Telerobot control via Internet, *Proceedings of the 1998 IEEE International Symposium on Industrial Electronics*, Bled, Slovenia, July 12-16 1999, Page 298-303.

De Queiroz, I., & Bergerman, G. (1998). *A robotic and computer vision virtual laboratory*. Proceedings of the 1998 International Workshop On Motion Control, Campinas SP, Brazil, 1998, Page 694-699.

Kuechlin, W., Gruhler, G., Lumpp, T, Speck, A., & Rupp, A. (1997.) *Highrobot: telerobotics in the Internet.* Proceedings of the 1997 IEEE 6th International Conference on Emerging Technologies and Factory Automation, Los Angeles, September 9-12 1997, Page 825-832.