“Operation of a robotic manipulator through the WWW”

Reinaldo A. C. Bianchi, Briand S. Moreira Jr., Fábio C. Ferraz and Anna H. R. C. Rillo.

Divisão de Automação e Inteligência Artificial - Laboratório de Sistemas Integráveis

Universidade de São Paulo (DAIA / LSI - EPUSP)

Av. Prof. Luciano Gualberto, travessa 3, 158 - 05508-900 São Paulo, SP, Brazil.

rbianchi-, briand- and fferraz@lsi.usp.br

Abstract

This paper describes a control system for a robotic manipulator that uses the WWW as the user interface, which is the first work towards the goal of building a virtual laboratory. It describes the robotic manipulator, the implementation of the interface that allows the submission of instructions to the manipulator, a specialized interface for a specific application and future works.

1. Introduction.

With the goal of sharing the equipment available in their laboratories with other institutions, several researchers are studying the virtual allocation of resources and the methods that can be used to allow researchers scattered around the planet to use their equipment. The name given to this kind of project is Virtual Laboratory.

Some of the existing projects aiming the development of virtual laboratories in research institutions around the world are:

- Distributed Collaboratory Experimental Environments (DCEE): at the Lawrence Berkley National Laboratory (LBNL), California, USA. [2]
- Worldwide Lab (WWL): at the Institut National de Recherche en Informatique et en Automatique (INRIA), Sophia-Antipolis, France.[5]
- ONIKA: at the Advanced Manipulators Laboratory (AML) at Carneguie-Mellon University (CMU), Pittsburgh, USA. [3]

Within this context, the Division of Automation and Artificial Intelligence at the Laboratory of Integrated Systemas (DAIA/LSI) of the São Paulo University intends to make its equipment available to other institutions. These equipment are those disposed in the DAIA's Flexible Assembly Cell [7].

Our goal at DAIA is to make the cell (manipulators, cameras and workstations, with its control software) available for operation through the WWW, creating a virtual laboratory based on the WWW. To achieve this, a project - named VirtuaLSI - is under development at DAIA. [4].

The final intention of this project is to provide the control of the cell using high level instructions, and applying Artificial Intelligence planning and computer vision systems to control it locally. A state representation of the cell wil also be built and presented to the remote user employing virtual reality methods and descriptions, like the VRML. The high level instructions can be based on GRAFCET models, a petri-net like description of processes, or a special purpose language.

The work described in this paper addresses the teleoperation of one of the cell manipulator through the WWW, a first stage into DAIA's goals. For this, was developed:

I. An interface that allows registered researchers with WWW access to send instructions to one of the cell manipulators. It is important that this interface allows only registered ones to work with the manipulator, because it controls who is going to use the cell, creating usage schedules and responsible users. This project is aimed for researchers to work, not to the WWW surfers to play with our equipment, because a wrong instruction can damage the whole cell.

II. A specialized interface, for a specific application that controls one of the manipulators.

It is also proposed here the construction of an interface that allows the reception of instructions sent by an application in a remote server. This is important because enables the remote user to develop his applications in a remote machine, interfacing with programs there, and only sending the instructions to the cell through the local server.

This goal has roots in a previous work [1], where the aim was to improve the autonomy of the assembly cell, introducing in it the AI Planner NONLIN [8]. In this work, the WWW was used as a local user interface due to the easiness of its implementation.

2. The Flexible Assembly Cell.

The Flexible Assembly Cell at DAIA [7] began to be built in 1989, in a combined work with the Institut fuer Prozessrechentechnik und Robotik (IPR) at Karlsruhe University in Germany.

Today the cell consists of: one workstation where the control system is based; two workstations for the vision system processing, one of which has a microcamera and image acquisition board; two Mitsubishi RM robotic manipulators with 5 degrees of freedom, each controlled by a PC microcomputer; a CCD camera with image acquisition board and a PC microcomputer. All the computers are linked by an ethernet local network. The cell has the configuration shown at figure 1, and a picture of it is shown at fig 2.

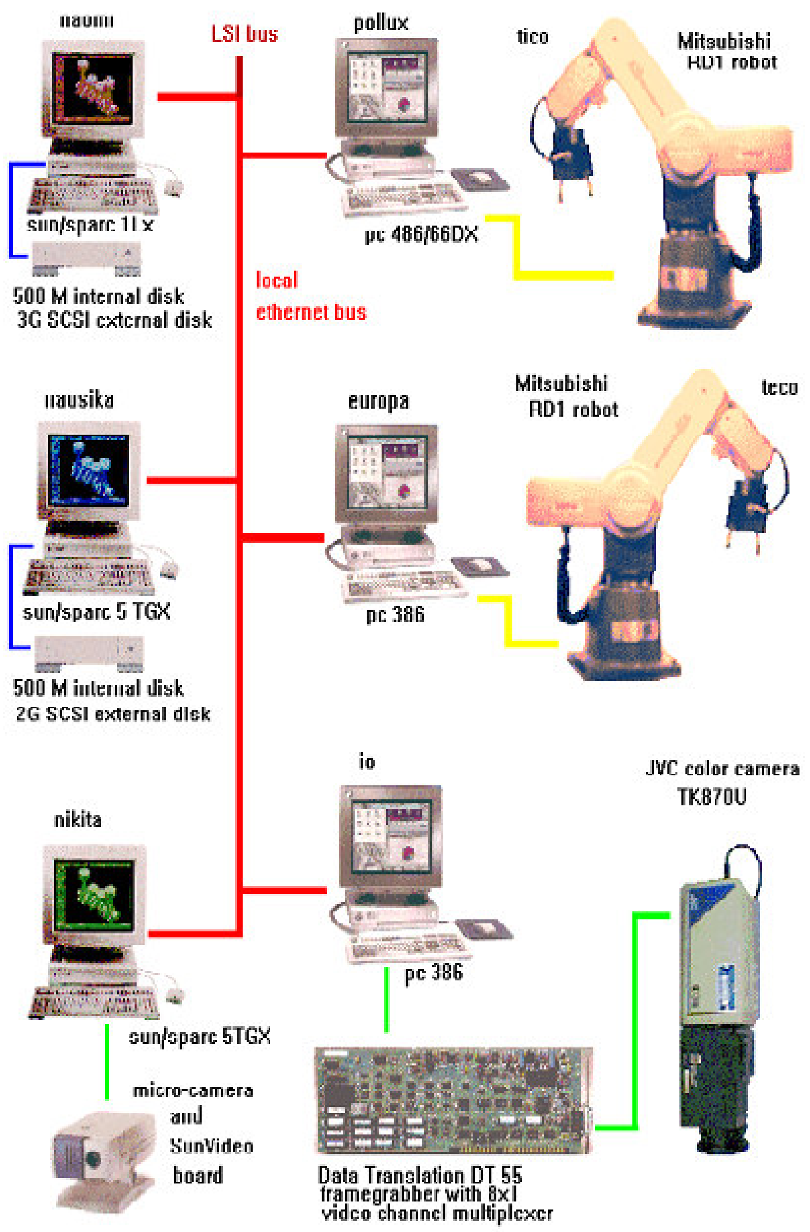


Fig. 1. The Flexible Assembly Cell configuration.

This cell can be used for small industrial assembling of small mechanic pieces, small toys, computer boards, separation and wrapping of products, quality control, etc. It is worth to note that the cell can be inserted in an assembly line and execute the assembling of one or more parts of a more complex product, not necessarily taking part in the assembly of the complete product.

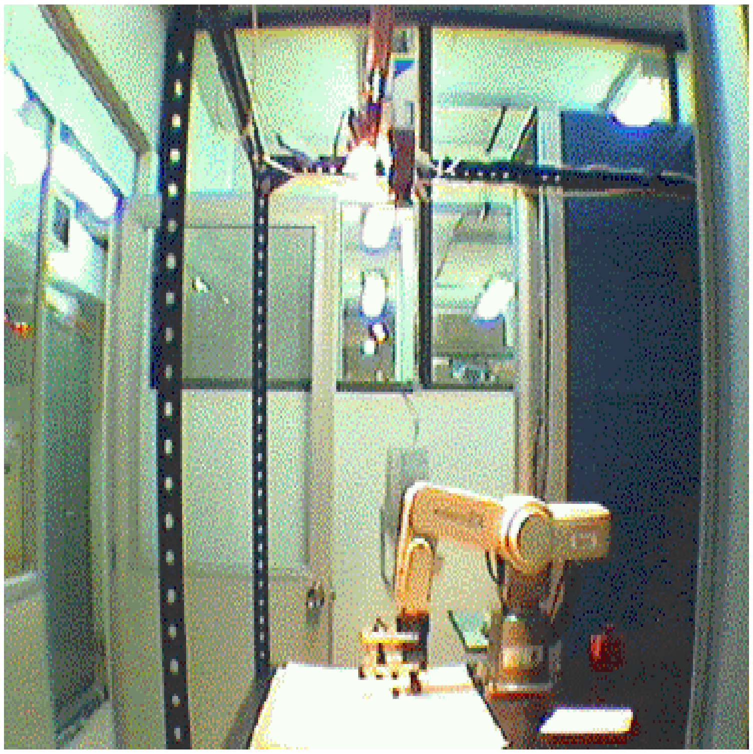


Fig. 2. The Flexible Assembling Cell - one of its manipulators.

3. The robotic manipulators.

The manipulators used are two Mitsubishi Movemaster vertically articulated, with 5 degrees of freedom, and a grip.

This manipulators have a variety of instructions that enables the control of their movements. This instructions can be classified according to their purposes in positioning instructions (like *move joint* and *move to position*, let the user specify the position of each joint or move the grip to a determined Cartesian position) grip control instructions (for the grip control), port and error instructions.

A microcomputer is needed to send the instructions to the manipulator through its serial communication port. There is a program in each microcomputer that controls the activity of each manipulator.

4. Teleoperation of one robotic manipulator.

A generic WWW interface for the control of the manipulator, in which instructions are sent directly to the manipulator is the first part of this work. The diagram in fig 3 shows the interface mechanism.

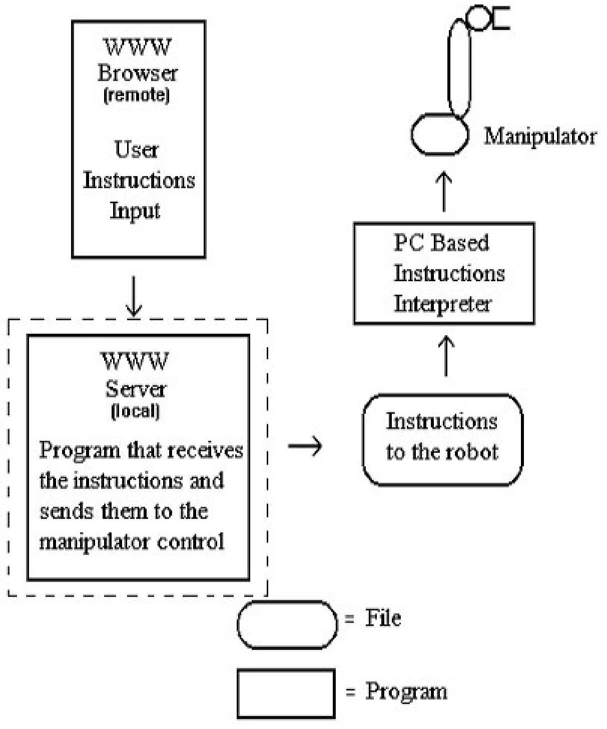

WWW
Browser
(remote)
User
Instructions
Input
WWW
Server
(local)
Program that receives the instructions and sends them to the manipulator control
Instructions to the robot
PC Based Instructions Interpreter
Manipulator
= File
= Program

Fig. 3. Mechanism of the interface.

When the URL *http://www.lsi.usp.br/~daia/celula/play./PLAY-ini* is requested by a browser, a program written in C in the local server checks if the cell is available. The utilization control is made checking a file at the server. If the cell is not available, due to the fact that its usage is shared with other projects, the server returns a document explaining why it is not available.

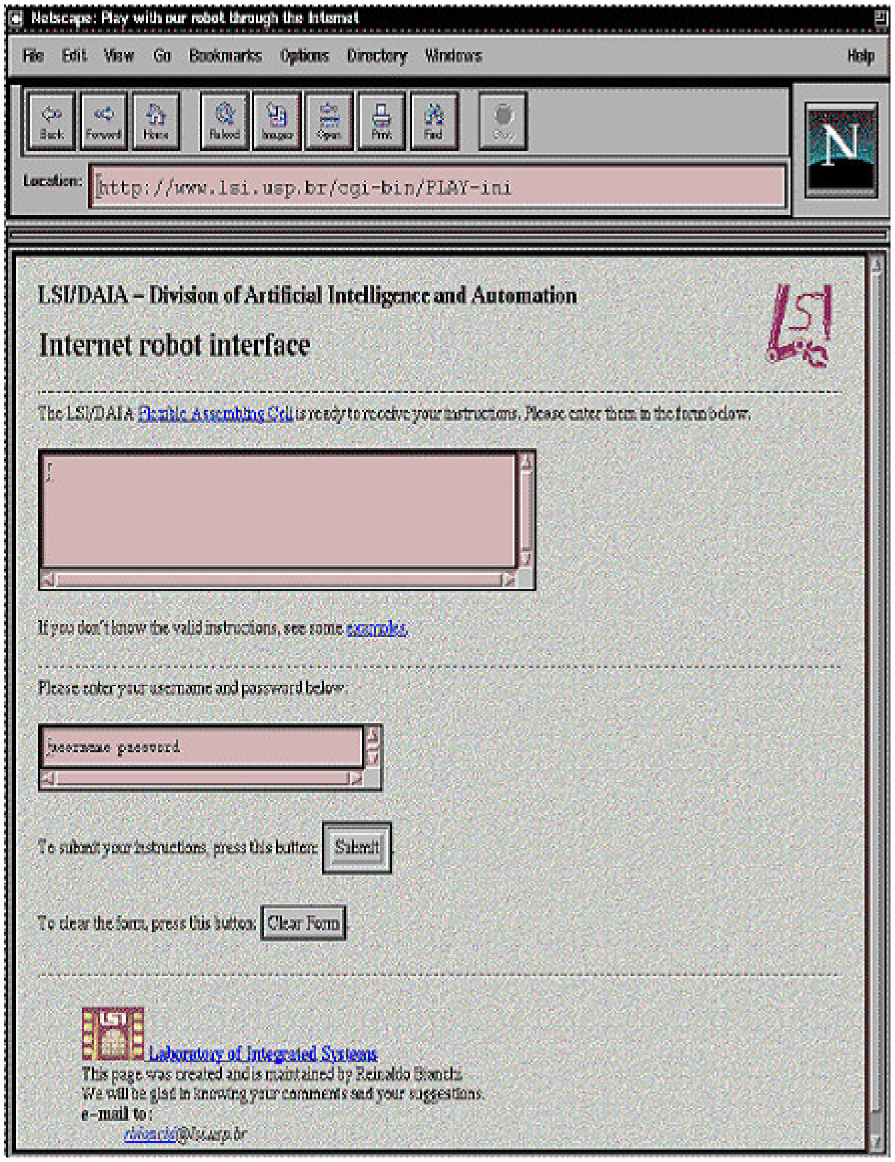
If the cell is available, the server returns a html document in which the user can input the instructions to the manipulator. This document has two form fields, one for the instructions and one for the username and password of the user. The instructions that can be submitted here are those of the manipulator.

After the form is filled, the client browser submits the data in its form fields to the local WWW server. When the server receives the data, it checks the user's password and, if it is a registered one, the instructions are written in a transfer readable for the PC that controls the manipulator.

A program is kept running in background at the PC, waiting for the transfer file to be created. Whenever the file exists, the program reads it and sends the instructions to the manipulator, accomplishing them. When it finishes sending all instructions, it deletes the transfer file.

The program in the WWW server keeps on running while the manipulator is in use. When the transfer file is deleted, it returns to the user a document with a status report. In the future, we also wish to return a video of the movement accomplished by the manipulator.

Today, the user can watch the manipulator working through a microcamera in a Sun workstation which is available at the cell. The user must run the SunVideo program on the machine with the microcamera, and the display is made at his machine, which needs to be working with OpenWindows. In the future, this will be also made with an inlined MPEG video displayed in a Netscape document.


Netscape: Play with our robot through the Internet
File Edit View Go Bookmarks Options Directory Windows Help
Back Forward Home Reload Images Open Print Find Stop
Location:
Submit
Clear Form

http://www.lsi.usp.br/cgi-bin/PLAY-ini

LSI/DAIA – Division of Artificial Intelligence and Automation

Internet robot interface

The LSI/DAIA Flexible Assembling Cell is ready to receive your instructions. Please enter them in the form below.

If you don't know the valid instructions, see some examples.

Please enter your username and password below:

username password

To submit your instructions, press this button:

To clear the form, press this button:

Laboratory of Integrated Systems
This page was created and is maintained by Reinaldo Bianchi
We will be glad in knowing your comments and your suggestions.
e-mail to:
rbianchi@lsi.usp.br

Fig. 4. The interface page in Netscape browser.

The document at the URL *http://www.lsi.usp.br/~daia/celula/play* has examples of the instructions that can be sent to the manipulator and instructions to the use of the interface.

5. An example of a specialized interface: tic-tac-toe.

5.1. Introduction.

The goal of the second part of this work is to build a specialized interface for a specific application for the cell. An interface to an application which allows a WWW user to play tic-tac-toe against a program that controls the manipulator was chosen, because game playing is an important field in AI research, one of the DAIA's main interest. Also, tic-tac-toe was chosen because it is a simple game, with known decision heuristic methods which are simple to be implemented. Another example of this kind of application is described in [1].

A board with three rows and columns and X's and O's pieces are located in front of the manipulator. Through the interface, the user makes his choices and a program at the WWW server computes the moves for the manipulator and accomplishes the movements of the pieces in the real board.

5.2. Sites related with tic-tac-toe in the WWW.

There were found plenty of sites related with the tic-tac-toe game in the WWW, some very good and others poor, judging for their interface and decision methods.

Most sites don't mention what is the algorithm used to compute the next move (though some sites allows the download of the source code of the program), so it was not possible to analyze the sites by their decision procedures. They were judged by their interfaces and process of creating the returning document.

Some of the sites found perform a combinatory explosion of the possible states. This means that for every choice the user can make, there is a previous prepared document, ready to be returned. Therefore, there is a "document database", with up to 1000 documents, waiting for the user decision to be sent. There is no real-time decision process, and all the decisions are hardwired. Some of these sites are: http://bourse.com/ttt/play.html, http://vpe.com:80/cgi-bin/vpecgi/tictactoe, http://zaphod.cs.uni-sb.de/cybye/tic/tic.html, http://www.epfl.ch/Staff/Yves.Piquet/clip2gif-home/tictactoe.cgi and http://fjwsys.lang.gov/cgi-bin/ttt.

Other sites use the WWW server to build in real time the return document, based on a decision method. At each time the user makes a move, a program in the server creates a return document with the user and the computer moves. This is a more efficient process which uses less memory and disk and allows more flexible games, with different difficulty levels. Some of these sites are: http://www.digicash.com/shops/games-dept/games-index.html, http://www.stardot.com/~lukessem/stubbed.html, http://www.eg.bucknell.edu/cgi-bin/ttt.

The use of the WWW server is important because, besides permitting the use of a flexible method to compute the next movement, it allows the control of the manipulator. Without the server, this is not possible.

5.3. The MINIMAX method.

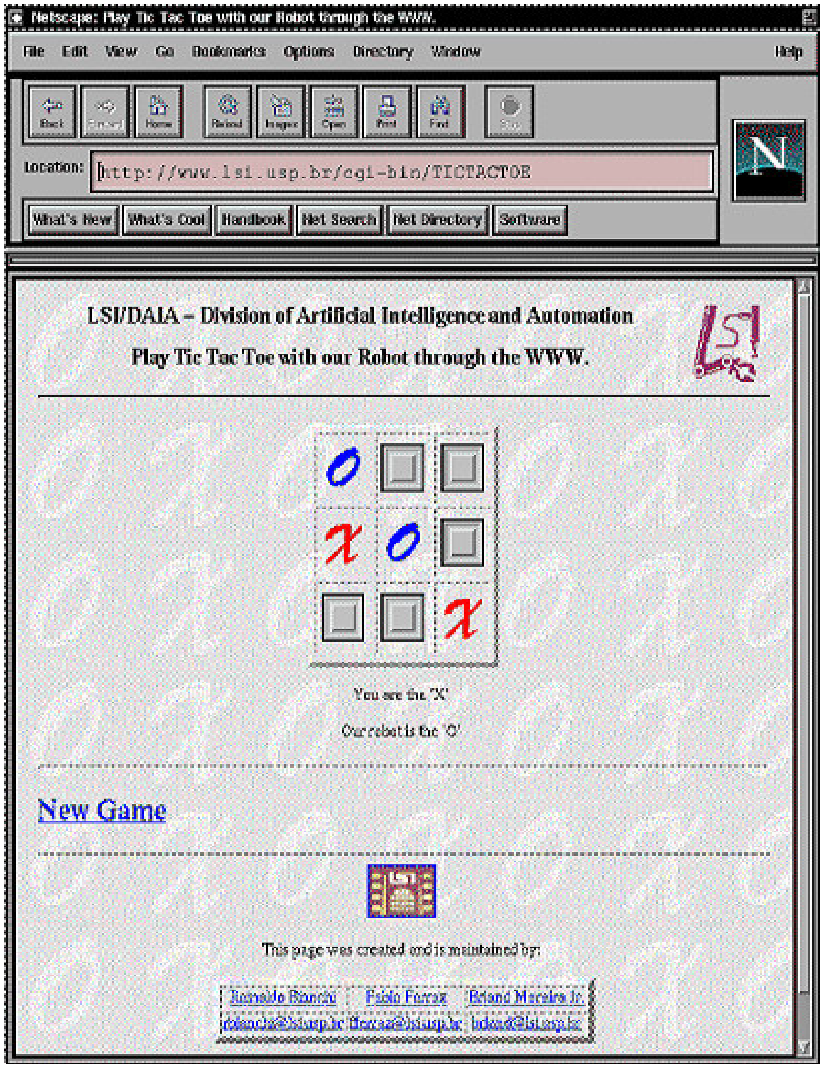
The decision procedure built is based on the MINIMAX search algorithm [6]. It receives a symbolic representation of the state of the board and decides what is the best move for the computer to make. To find that, it executes a depth search in a tree of possible states. The algorithm allows three different levels in which the tree can be expanded, the depth of the search reflecting the forecast the algorithm makes of the next moves. Therefore, there is three difficulty levels, easy medium and hard, reflecting one, two or three depth levels in the search tree.

The algorithm is based on the premise that to win the game both players always try to make the best move at each turn. It uses an heuristics to determine what is the best move, based on the chances of victory in the lines, columns and diagonals for both players, and returns the best option at each time.

5.4. Implementation.

The construction and working method of this interface is similar to the interface first described in the previous section. The difference is that this one does not only receive the instructions and send them to the manipulator, but it uses higher level input data and compute the movements needed for the manipulator. Also, this kind of interface allows both manipulators to be controlled by the application, and its integration with other software.

When the URL*http://www.lsi.usp.br/~daia/celula/tictactoe/* (fig 5.) is requested by a browser, a program in the local server checks if the cell is available. The utilization control is made exactly as in the interface described in section 3. If the cell is not available, the server returns a document which let the user to play tic-tac-toe, but which does not reflects the game at the cell. If the cell is available, the server returns a html document with the initial board. This program also resets a file that contains the description of the board.


Netscape: Play Tic Tac Toe with our Robot through the WWW.
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
What's New What's Cool Handbook Net Search Net Directory Software

Location: http://www.lsi.usp.br/cgi-bin/TICTACTOE

LSI/DAIA - Division of Artificial Intelligence and Automation

Play Tic Tac Toe with our Robot through the WWW.

You are the 'X'

Our robot is the 'O'

New Game

This page was created and is maintained by:

Reinaldo Bianchi	Fabio Ferraz	Erland Moreira Jr.
rbianchi@lsi.usp.br	fferraz@lsi.usp.br	[illegible]@lsi.usp.br

Fig. 5. The interface page in Netscape browser.

The board in the html document is composed of a table with 3 rows and 3 columns of button, that the user can click one. When a button is clicked, the browser submits which button was clicked to the server, starting a program that executes the following tasks:

- Receives the button chose by the human opponent.
- Loads a file which contains the description of the board.
- Sends the instructions of the human user's movement.
- Waits for the manipulator to accomplish the instructions sent.
- Checks if the user won the game. If he did, returns a document acknowledging that.
- Computes a move using the MINIMAX method.
- Sends the instructions of the computer movement.
- Waits for the manipulator to accomplish the instructions sent.
- Checks if the program won the game. If it did, returns a document informing that.
- Writes the file with the new configuration of the board.
- Returns a document with the new configuration of the board.
- Halts.

The submission of instructions to the manipulator controller computer is done in the same manner as in the previous section, and it is also written in C. In the future we intend to include in the document an inlined MPEG video showing the manipulator moving around in real time. The diagram in fig 6 shows the interface mechanism. Note that the difference to fig 3 is only related to the WWW server. It is worth remarking that at each turn of the human player, a new interaction with the server is made. The program restarts at each time. This is why we need to keep the board configuration in a file.

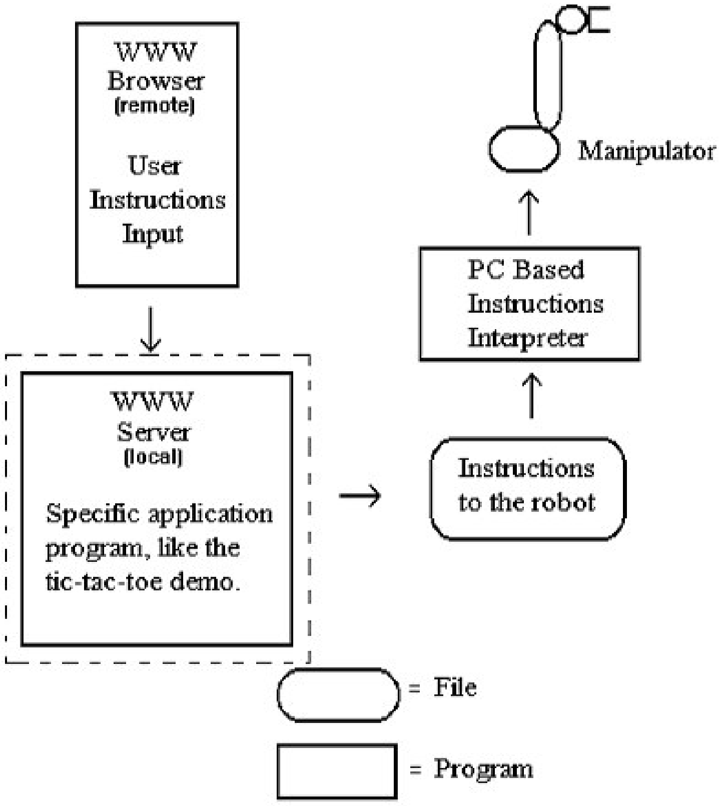

WWW Browser (remote)
User Instructions Input
WWW Server (local)
Specific application program, like the tic-tac-toe demo.
Instructions to the robot
PC Based Instructions Interpreter
Manipulator
= File
= Program

Fig. 6. Mechanism of the application.

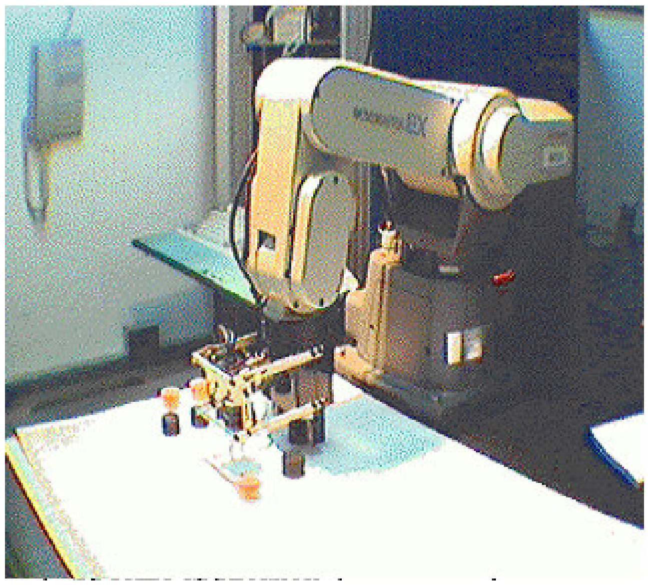


Fig. 7. The robotic manipulator and the pieces.

The document at the URL *http://www.lsi.usp.br/~daia/celula/tictactoe/* explains this application and allows the choice of the level the user wants to play. From it you can start the game.

6. Applications based on a remote server.

The possibility to create applications as presented in the previous section is a second step towards the creation of our virtual laboratory, because until now the remote user cannot create his own application and submit the instructions from a remote server to the manipulators. As already explained, this is important because allows the remote user to develop his applications in a remote machine, interfacing with programs located there, and only having to send the instructions to the cell trough the local server.

To create this kind of application, a program managing the arrival of instructions sent by programs in remote WWW servers is needed. This program receives the instructions and then forward them to the manipulator computer. The diagram in fig 8 shows the interface mechanism. Again, the difference to fig 3 and 6 are related only to the WWW server. Figure 7 shows the manipulator and some pieces.

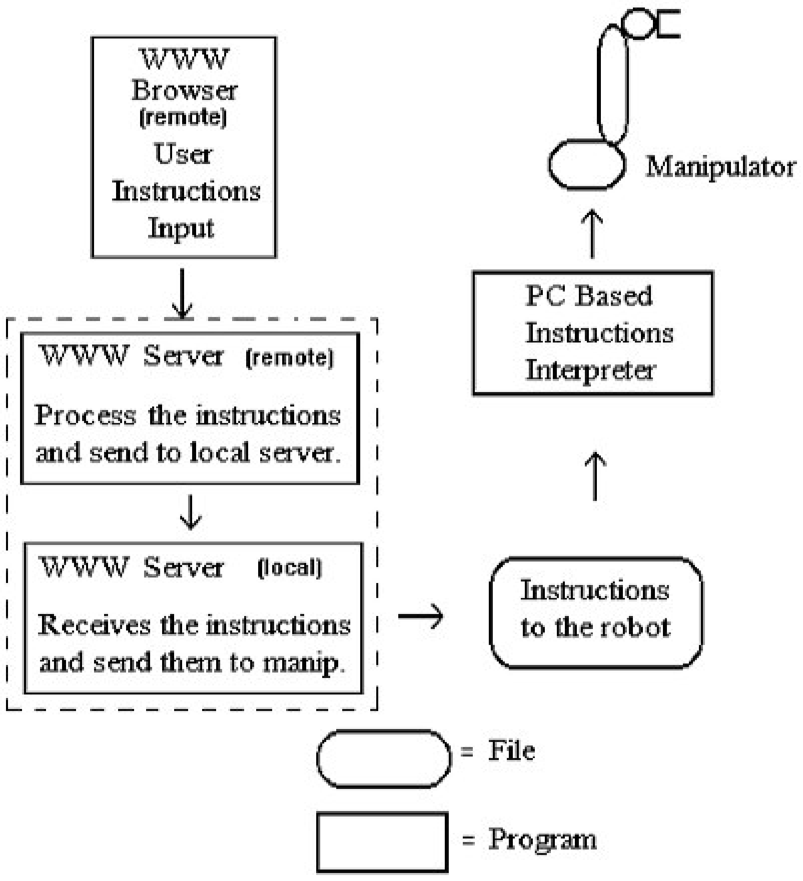

WWW Browser (remote) User Instructions Input
WWW Server (remote)
Process the instructions and send to local server.
WWW Server (local)
Receives the instructions and send them to manip.
Instructions to the robot
PC Based Instructions Interpreter
Manipulator
= File
= Program

Fig. 8. Mechanism of the interface.

This work is not concluded yet, and is based in the exchange of messages between the two servers. Also, this program has to verify the authenticity of the messages, for the same reasons we have passwords for the interface described at section 3.

Finally, this kind of interface also allows the control of the cell as a whole, while the application at the remote server can interact with other programs.

7. Conclusion and future work.

This paper describes the initial steps for the creation of virtual laboratories using the WWW as the base of communication, work that enables us share the available equipment at the DAIA/LSI with other institutions. It presents the basis of this work in a specific interface and in a specialized interface, describing how they work and how they were implemented.

In a short term, an important improvement to the user interface is to show the manipulator movements in a inlined MPEG video in a WWW document. A soon as the video capture program is available this will be implemented. Today, to watch the movements the user must have access to the machines at the laboratory to run the SunVideo program that allows the display of the image at a remote workstation.

Because of the risks to our equipment we developed a way to check the authenticity of the submitted instructions, but still there is the need of submission of accurate instructions by the researchers. A future work is to create a program that checks the instructions to find if they will create dangerous situations, and avoid them, warning the user and stopping the manipulators movements.

This work contributes to a wider goal of building a global research community, showing a promising way to create a virtual laboratory.

Finally, the interface described in section 5 - which receives instructions from an application located in a remote WWW server -, is still to be completed. The main problem to solve is the transfer of messages between two servers.

Acknowledgments

We'd like to thanks João Kogler Jr., for the figure 1 and the project VirtuaLSI, from where the information about the virtual laboratories porjects existing was taken. we also thank Katsumi, Mikael, Gélio and Tuca for the constant help and for programs that aided us in this work. Finally, Márcio and Anna Rillo, our advisors.

References.

[1] Bianchi, R. A. C.; Santos, M. V. T.; Rillo, M. "Integration of an AI planning system in a flexible assembling cell". In: BRAZILIAN SYMPOSIUM ON INTELLIGENT AUTOMATION, 2., Curitiba, 1995. **Proceedings.** Curitiba, SBA, 1995. p.195-200.

[2] Chew,J. "The distributed , collaboratory experiment environments (DCEE) Program" , www html document , http://www-itg.lbl.gov/DCEEpage/DCEE_Overview.html , version Sep. 25 , 1995.

[3] Gertz,M.W. , Stewart,D.B. , Khosla,P.K. "A human-machine interface for distributed virtual laboratories" , IEEE Robotics and Automation Magazine , Dec. 1994 , pp. 5-13.

[4] Kogler Jr., J.E. et al "Towards the implementation of a virtual distributed laboratory at the LSI - USP" . EPUSP/PEE/LSI Technical report - in preparation. 1996.

[5] Mangin,F. "Join the World Wide Lab project", www html document , http://wwlab.inria.fr , version Jul. 1995.

[6] Rich, E. **Artificial Intelligence.** New York, McGraw Hill, 1983.

[7] Rillo, M.; Rillo, A.H.R.C.; Costa, L.A.R. LSI assembly cell. In: IFAC/IFIP/IFORS/ IMACS/ISPE Symposium on Information Control Problems In Manufacturing Technology, 7th, Toronto, Canada, 1992. **Proceedings.** Toronto, Canada, IFAC, 1992.

[8] Tate, A. "Generating project networks" In : INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 3, 1977. **Proceedings**. 1977. p.888-9.