PLAYING TIC-TAC-TOE WITH ROBOT ON THE WEB IN REAL TIME

K. Yeung[1], B. Jacques[2] and R. Du[3]

Abstract *¾ This paper presents an interactive real-time man versus robot playing "Tic-Tac-Toe" on the Internet. The game can be accessed from* http://emlserv.acae.cuhk.edu.hk/. *Anyone in the world can play the game against the robot and see the results in real time. Our system consists of a web server, a video server, an Artificial Intelligence (AI) rule base running on the web server machine, and a robot with digital I/O. The key to the system includes a reliable and robust computer network system structure and the AI game playing strategy. According to our test, the system is reliable and smart (i.e., the robot will not loss to human players).*

Index Terms ¾ TIC-TAC-TOE, robot, Internet, AI.

INTRODUCTION

Teleoperation, in particularly telerobotics operation, through Internet is one of the most active R&D topics today. A large number of research projects have been carried out on how a human operator can conduct a task in a remote environment over the communication network. As a result, tremendous progresses have been made. To provide more interactive functions, sensing and message feedbacks are also included. One example is found in UC-Berkeley [1], which allows a group of users to teleoperate an industrial robot arm simultaneously. This is one of the first collaboratively control robots on the Internet. The other example is the network-based robotic system that uses Common Object Request Broker Architecture (CORBA) to implement networking connections between a client and a remote system. The system is located at the University of Electro-Communications, Japan [2]. It allows a client to invoke an operation on a server across network transparently without needing to know where the application servers were located, or what programming language and operating system are used. At the Chinese University of Hong Kong, a web-based control system was successfully developed, which allows clients conduct automatic control experiments like DC motor control and water-coupled tank control in real time through Internet [3]. An innovative real-time remote-access control engineering teaching laboratory was developed and demonstrated at Oregon State University in 1998 [4]. The system "Second Best to Being There (SBBT)" provided a Java interface on the client's browser to conduct the robot experiment. A common undergraduate physics experiment was modified for remote operation over the Internet using LabVIEW programming environment at James Madison University [5]. However, there are a number of bottleneck problems remain unsolved. For example, the signal transmission delay often causes a great barrier to achieve fast and reliable teleoperation. In other to solve this problem, a concurrent bilateral teleoperation over Internet from Pusan National University in Korea, utilized predictive displays by providing the simulated or predictive feedback [6].

In this paper, a real time intelligent robotics game is introduced. The game demonstrates the telerobotics operation in a fun game-playing manner. It also helps students to learn more about robotics and AI on the Internet on-line any time and anywhere.

The rest of this paper is organized as follows: Section 2 describes the architecture of the system as well as the hardware and software configurations. The interface is described in Section 3. Section 4 describes the model and strategy of playing the game "Tic-Tac-Toe". An playing example is given as well. Finally, Section 5 offers conclusions.

SYSTEM CONFIGURATION

Figure 1 shows our system setup. A 5-DOF PERFORMER-MK3 robot is used to carry out the physical actions (i.e., playing Tic-Tac-Toe). It has two communication channels with a PC computer: one is through a digital I/O board to receive commands (e.g., move to specific location) and to send responds (e.g., an action is completed). The other one is through a RS232 port for downloading robot programs. It is used only when the robot program needs a change. A real-time camera with a video server broadcast the video signal onto Internet. Finally, a computer server provides the web content and stores the data. It runs two programs: one is a CGI program that handles the data transmitted from clients' web browser. The second one is a PERL program that carries out the game playing AI logic. It also sends the control signals to the PC computer through TCP/IP.

The system can be divided into four modules: the web server module, the robot and controller module, the video server module and the digital signal generator module. Each module functions independently and yet they are closely linked to make the whole system work. Such a design allows the system being modified easily. For example, we can

[1] Please send the corresponds to this author at kyeung@acae.cuhk.edu.hk
[2] Ms. Brenda Jacques was a senior student at the University of Windsor in 1997, who developed the gaming strategy as a part of her senior project under the supervision of Dr. R. Du.
[3] R. Du, The Chinese University of Hong Kong, Shatin, Hong Kong, rdu@acae.cuhk.edu.hk

replace the Tic-Tac-Toe game with a chess game without structural changes.

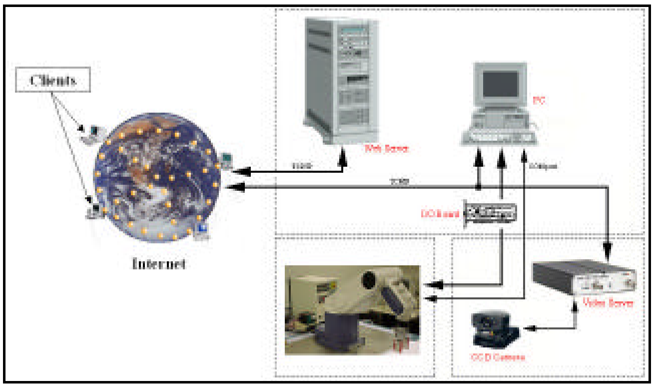

FIGURE. 1
ILLUSTRATION OF SYSTEM HARDWARE CONPONENTS

Figure 2 depicts the software architecture of the system. The server uses HTML, PERL, and JavaScript. While LabVIEW's G language is the one used on the PC side. LabVIEW from *National Instruments* is used to act as the TCP communication server with the web server and digital signals acquisition and broadcasting interface. All networking algorithms are based on client/server structure. TCP is the primary protocol used in handling communications.

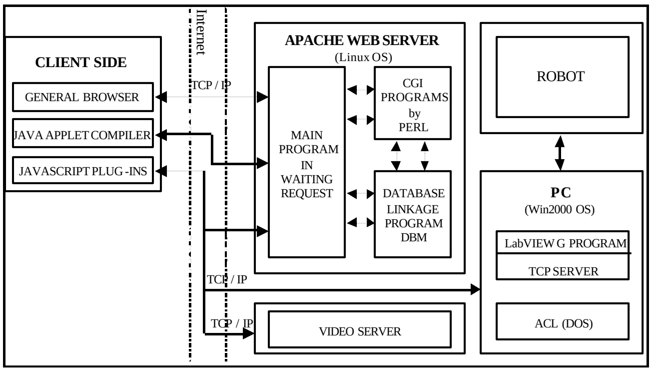

FIGURE. 2
SOFTWARE STRUCTURE

We use Apache web server on Red Hat 7.1 Linux operating system to provide web pages, user and viewer authorization and user conflict checking. According to the Survey by Netcraft [7], 62.55% of the users are Apache Web Server user's. The highly stable performance of Linux operating system can greatly enhance the stability of the overall system.

All CGI in our web server are written in PERL language, which is the most popular and powerful way to handle CGI today. CGI can be coded by using different languages, such as PERL, C, and JAVA, etc. Note that CGI is neither a program nor a script. It is a standard for setting up interaction between external application and information servers, such as web servers. Web servers can do more than just sever static pages. They can execute scripts so that more dynamic information can be shown. To create consistent scripts that work on any web servers, the standard script interface, CGI, is used. CGI is simple to implement, portable, and completely transparent to the end user.

Program language PERL processes user input and returning HTML text pages. We use PERL because of its natural fit for this new programming environment. PERL is a GNU open-source programming language whose syntax is a cross between the programming and scripting languages C/C++, Awk, and Sed. PERL is a great choice for an all-purpose scripting language, thanks to its one uniform base syntax across all platforms and its relatively quick, single-step, compile-and-execute structure [8]. Perhaps the most complicated part of developing advanced web functions is implementing back-end processes, such as capturing form input or upload files, accessing databases, and automating system administration functions. HTML does not make decision nor execute to response to client's willing. Thus, in general, back-end processing can run as a CGI script.

User authorization and conflict checking as well as database linkage are all handled by PERL language on the server. Its library packages on database management system (DBMS) provide modules to interface with Oracle, Sybase, mSQL, MySQL, Ingres, and others [9]. We use DB_File packages to conduct database management. For a better communication and error tracking, Web Server keeps an event log database with information on the system status, current user's login and logout time. While the game is playing, robot's status of either moving or stationary keeps updating in our database as well as it is so important to those real-time control issue.

The robot is controlled by its own program ACL (Advanced Control Language), which is burned onto a set of EPROMs in the Controller-AC. ACL can be accessed from any standard terminal or PC computer by means of an RS232 communication channel.

WEB INTERFACE

As shown in Figure 3, while a user is authorized to active the system (no body is playing at this moment), he/she can start the game on a web browser. The real-time video window is located at left-top corner and the user can control the camera by zoom/pan/tilt mode. The playing board is on left-bottom one with 9 boxes, the user always uses a green 'cross (X)' and robot uses a red 'naught (O)'. The user can choose the restart or logout the game while it is in his/her turn (robot stopped). The message feedback bar, which gives the feedback and shows system's status, is put on right-bottom corner. Some notes are offered on the right side of the page.

One may also view the game (some body is already occupied the system), he/she will get the similar graphical interface, but cannot do anything else. There will be no camera control. The System will automatically update the game's situation.

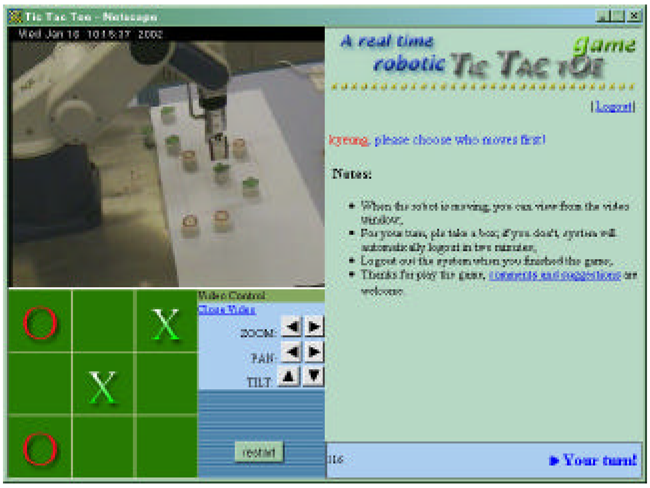

FIGURE. 3
SYSTEM WEB INTERFACE

THE AI GAMING STRATEGY

Definition of the Game

Computer gaming refers to the computer's ability to "*serve as an intelligent opponent, even to stand in for a second human game player*" [10]. Game playing has been the primary source of evaluating the quality and progression of artificially intelligence. Tic-Tac-Toe is a simple game that its simplicity will aid in the exploration of the computer decision making process in order to mimic the human player. Finite numbers of outcomes, and the absence of chance, provide an easy model for play the game. The premise of Tic-Tac-Toe consists of a board with nine boxes such as shown in Figure 4.

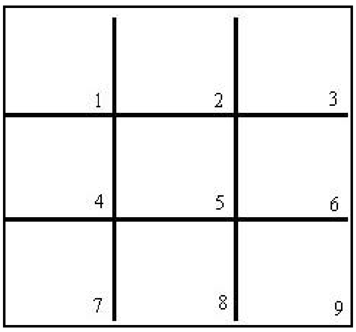

FIGURE. 4
GAME BOARD

The object of the game is to achieve a 'win'. A 'win' consists of three boxes in a straight row that can either be horizontal, vertical or diagonal, and all being of the same sign. Each player is allocated a sign, either a cross 'X' or and naught 'O' which they will mark on the game board box during their turn. Alternating turns, each player makes their move by marking their sign in one of the nine boxes until one of the players wins. If all the boxes are occupied and neither a 'win' nor 'loss' has occurred, then a 'tie' occurs. The outcome of the game is deterministic, meaning that it results in one of three possible outcomes namely, a 'win', a 'loss', or a 'tie'. Each box can be described as being occupied by a 'cross', a 'naught', or being 'empty'.

The Model

In order to develop an intelligent Tic-Tac-Toe gaming strategy, a mathematical model is used to define the state of the game. For simplicity, the computer was always naught 'O' and the opponent in turn was always cross 'X'. A numerical value was assigned to each sign. An 'O' in this model has the equivalent value of 10 and 'X' has the value of -1. Based on the summation of the rows the computer will determine its next strategic move.

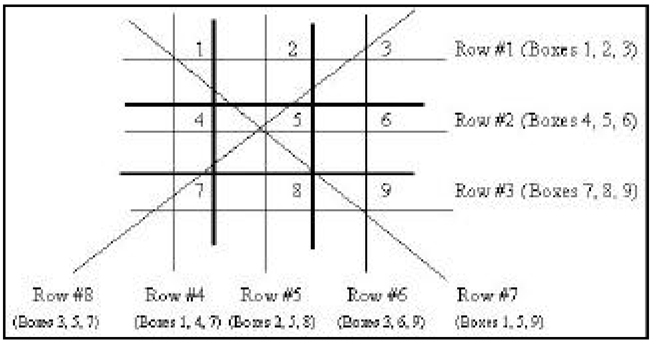

FIGURE. 5
ROW DEFINITION

The game states are defined as in Figure 5. All possible row combinations and their corresponding row values are summarized in Table 1. Note that the order itself is irrelevant since the summation of the row values are the same regardless.

TABLE I
ROW COMBINATIONS AND ITS EQUIVALENT VALUE

Row Number	Row Combinations	Row Equivalent Value
1	(, ,)	0
2	(O , ,)	10
3	(O , O ,)	20
4	(O , O , O)	30
5	(X , ,)	-1
6	(X , X ,)	-2
7	(X , X , X)	-3
8	(X , O ,)	9
9	(X , X , O)	8
10	(O , O , X)	19

Strategy

The objective of the computer is defined as the process of creating a 'winning' condition continuously scan the board conditions in an attempt to block all 'potential winning' conditions that its opponent may present. Logic states are defined and associated with their appropriate row summations values, to represent the 'winning' and/or 'losing' potentials. Under current state, computer computes the sum of all possible next state, and then decision is made based on the rules shown below:

State 1 (O, O, O): "Has the computer won"?
State 2 (X, X, X): "Has the opponent won"?
State 3 (O, O,): "Can the computer make a winning move"?
State 4 (X, X,): "Can the computer block a possible win for the opponent"?
State 5 (X, ,): "Can the computer block opponent's potential win-win condition"?
State 6 (O, ,): "Can the computer make a move to increase its chance to win"?
State 7 (O, X,): "Last move state" ?

An Example

Now let us use an example to show the steps how robot responses to user's inputs. As illustrated in Figure 6, where robot is made to move first, a red naught 'O' represents robot's turn – Step 1 (by Rule 6 above). Then the user chooses the 7th position and puts the cross 'X' here for his/her move – Step 2. Now the computer starts to compute all possible row equivalent value. It gets the highest value of 20 (Table II) if it puts the naught 'O' in the 1st position – Step 3. Similarly, refer to Figure 6 Step 1 to 9, user and robot move respectively with strategy to block opponent's way to achieve a 'win'. The game ends tie.

TABLE II
MAXIMUM ROW EQUIVALENT VALUE FOR EACH POSITION PUT IN STEP 3

Position to Be Put (See Figure 4 for Position)	**Maximum Row Equivalent Value** (See Figure 5 for Row Definition)
1	20 (Row #7)
2	20 (Row #5)
3	19 (Row #8)
4	20 (Row #2)
6	20 (Row #2)
8	20 (Row #5)
9	20 (Row #7)

A number of problems could be raised while the application is set to operate by a robot in real-time. During a real-time process, particularly on telerobotics, system must be able to handle all possible conditions presented; otherwise there may be even more unpredictable events may occur. Problems as the same is played on the web platform, where the stability is presented concern.

Some more possible and abnormal cases should be considered include:

- User close the web window violently without finishing the game;
- User logout the game but the signals fail to transmit to server due to network traffic;
- User restart the game, which is not yet finished;
- Server fails to send the signals to the PC TCP server.

Thus, in order to ensure the robustness of system flow control, we keep on tracking the system continuously. Each time when it performs an action, status parameters are loaded and based on the status the correct response is given; then update the parameters again. For example, if we found that there is no user playing (user just logout), but the robot is still moving, then it must be the state that the robot is clean the board and not yet finish, so the system ask for waiting and displaying on message board. The system flow chart is shown on Figure 7 below.

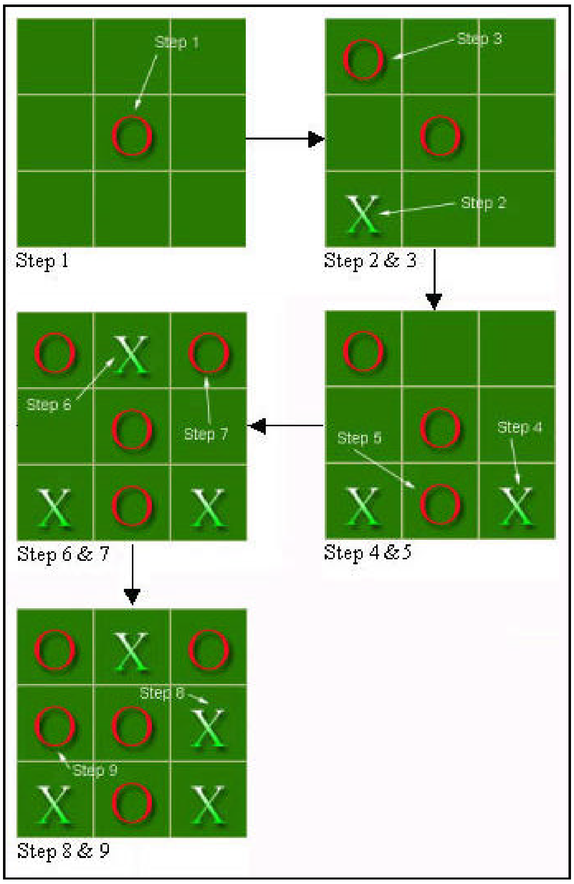

FIGURE. 6
STEPS ON GAME BOARD FOR ROBOT VERSUS MAN

It is challenging to handle and detect all the possible cases and errors. Nevertheless, our design seems minimize the possible error.

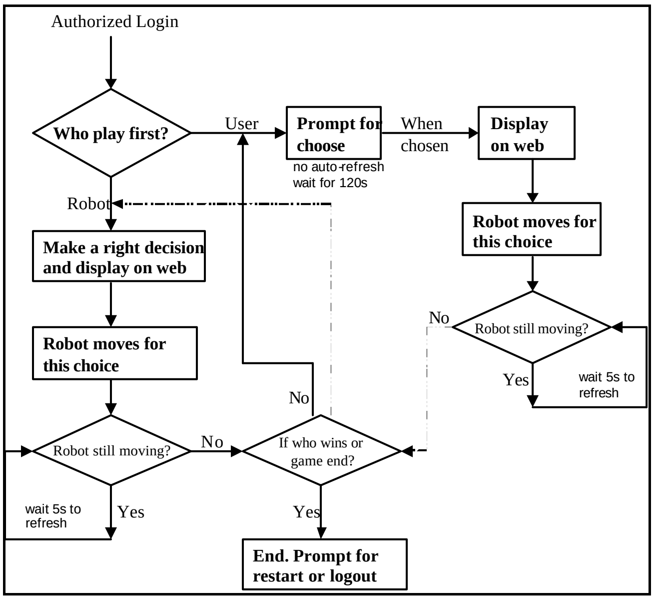

FIGURE. 7
SYSTEM FLOW CHART ON WEB SERVER SIDE

CONCLUSION

This paper presents a real-time web-based interactive tic-tac-toe of man versus robot game. Its implementation is a balance of web-programs, robotics, and AI. System hardware and software architectures are integrated to generate such a robust system. Anyone in the world can play and watch the robot in real-time through Internet. Video and text feedback are provided. The work helps not only on web-based robotics, but also on distance education.

REFERENCES

[1] Ken Goldberg, Billy Chen, Rory Solomon, Steve Bui, Bobak Farzin, Jacob Heitler, Derek Poon, Gordan Smith, "Collaborative Teleoperation via the Internet", *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, San Francisco, CA, April 2000, pp. 2019-2024.

[2] Songmin Jia and Kunikatsu Takase, "Network-Based Robotic System Using CORBA as Communication Architecture", Japan.

[3] Kin Yeung and Jie Huang, "Development of the Internet based control experiment", *40th IEEE Conference on Decision and Control*, Orlando, Florida, Dec 2001, pp.2809-2814.

[4] A. Bhandari, M.H. Shor, "Access To An Instructional Control Laboratory Experiment Through The World Wide Web", *Proc. Of the 1998 American Control Conference,* Philadelphia, 1998, pp. 1319-1325.

[5] C. Lon Enloe, William A. Pakula, Greg A. Finney, and Ryan K. Haaland, "Teleoperation in the Undergraduate Physics Laboratory—Teaching an old Dog New Tricks", *IEEE Transactions on Education*, Vol. 42, No. 3, August 1999.

[6] Jin-Woo Park and Jang-Myung Lee, "Concurrent Bilateral Teleoperation over the Internet", *Proceedings of ISIE 2001 IEEE International Symposium*, vol.1, 2001, pp. 302 –307.

[7] Netcraft Web Server Survey, http://www.netcraft.com/survey.

[8] Dennis R. Copeland, Raymond C. Corbo, Susan A. Falkenthal, James L. Fisher, and Mark N. Sandler, "Which Web Development Tool is Right for You?" *IT Pro*, IEEE, March | April 2000.

[9] Tom Christiansen & Nathan Torkington, "Perl Cookbook", O'REILLY, 1999.

[10] Artificial Intelligent, and Robot Wisdom, http://www.mcs.net/~jorn/html/AI/simiutation.html.