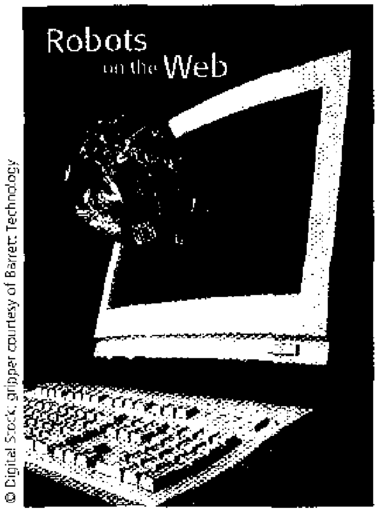

KhepOnTheWeb: Open Access to a Mobile Robot on the Internet

Studying One Year of Usage Data to Understand Current Technology, User Reaction, and Equipment Requirements for Remote-Control Web Robots

by PATRICK SAUCY and FRANCESCO MONDADA

For years research has focussed on ways to allow remote access via standard communication networks to unique or expensive structures such as supercomputers, important databases, unique information resources, the World Wide Web, e-mail servers, and ftp servers full of software. With the growth of the Internet, one finds more and more devices such as coffee machines, cameras, telescopes, manipulators, and mobile robots connected to it. Despite the fact that one may spy on other people with hundreds of cameras, it is currently possible to interact only with a few robots, which often have restricted access [1].

There are several explanations as why this is so. An installation with a robot is very expensive, regular maintenance is needed, and cameras are more user-friendly. To use a camera over the web, the user usually just sits and watches or sometimes has the ability to choose different camera orientation/views [7]. With a robot, you have strong interaction. For instance, with a mobile robot equipped with an arm you can move along the floor and grasp objects [8]. Discovering the control interface, the user has to understand rapidly the goal of the site and what the possibilities of the robot are in order to achieve them. A very famous example is the "Mercury Project" [2]. This kind of experiment is very useful in the sense that it gives important information about the reactions of the users, the kind of equipment needed, and the constraints of the Internet. More information about devices connected to the net can be found in [9] and [10].

This article analyses one year of netsurfer behavior regarding the use of KhepOnTheWeb, which was realized to demonstrate some possibilities of remote control of a Khepera mobile robot. After one year of access, we performed an analysis of the log files in order to understand the behavior of the public facing such an installation. This analysis was rather difficult because of the large amount of data involved, and specific software was developed in order to extract and present the relevant information. The goal of the project is presented, the hardware and software components of our robot installation are described, and the analysis of the web server log files is discussed. We also introduce another concept of a remote-controlled robot on the web.

Project Goal

The long-term goal of our project, which is called "Sharing of Unique or Expensive Equipment for Research and Education" (part of the Swiss Priority Program ICS of the Swiss National Research Foundation) is very similar to those of the experiments described in the introduction; namely, to provide access, through network communication facilities, to a complex and unique mobile robotics setup. The specificity of our project is that we want to provide a setup mainly for the scientific community for carrying out research in mobile robotics control.

The goal of the project fits well with the activity of our lab. The LAMI (Microprocessor and Interface Lab) is specialized in the development of mobile robots and tools for control algorithms research. The remote experimentation is the logical extension of this activity.

In this article we describe the results of the first part of this project, aimed to understand the possibilities of the current technology, the reactions of users, and the requirements for

1070-9932/00/$10.00©2000IEEE

the equipment of such a setup. In comparison with other projects on the web, our installation has some additional features:

- The controlled device is a mobile robot equipped with an on-board camera. This is also the case of some other setups that are partially available [4].
- Unlike other mobile robot setups, ours is running daily without the need of external support.
- The interface has a live video feedback and runs in a standard web environment.
- Everyone has access to our robot. There is no distinction such as a registered user/guest. But only one person at a time can control the robot, for a maximum of 5 minutes.

We had 27,498 visits performed by 18,408 unique machines. Only 3,178 machines did more than one access.

System Architecture

Our setup consists of a mobile robot accessible via the Internet that moves in a wooden maze (see Fig. 1). The labyrinth is made in such a way that the visitor can see various effects. In a general way, the walls are higher than the robot, so that the visitor has to move around to explore the maze. Only some center walls are lower than the others. On the right there is a mirror allowing the user to watch the robot that he is controlling. On the bottom left of Fig. 1 there is a ramp that gives the robot access to a view outside the maze.

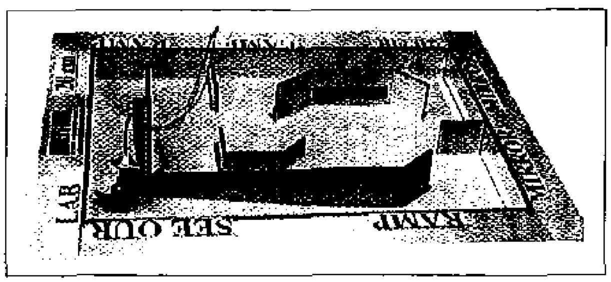

Figure 1. Khepera is in the lower left-hand corner of the maze, with its on-board video camera looking outside of the maze. The maze is 65 × 90 cm.

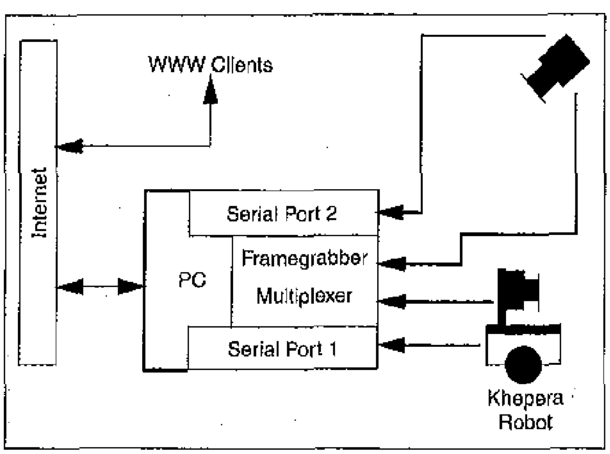

Figure 2. Connections between the devices of the setup.

In this experiment we use the mobile robot Khepera, which was developed in our lab in collaboration with K-Team, the company producing and selling it. Khepera is equipped with an on-board CCD camera and is connected to a PC via a wired link. The camera sends the video signal to a framegrabber that is also placed in the same PC. This computer is also our web server and is therefore connected to Internet. The user can access this installation via the Netscape web browser. Internet Explorer is not supported at this time.

A virtual model of the setup has also been developed using VRML and Java. This provides the user with the possibility to train himself by running a simulation locally on his machine, without time-delay constraints. A complete description of this implementation can be found in [5].

The Khepera Mobile Robot

Khepera is a small cylindrical robot, 55 mm in diameter and of variable height. Its small size allows us to employ a suspended cable for power supply and other signals without disturbing its movements. A camera observing the environment can be placed without the need for wide-angle lenses. In the configuration shown in Fig. 1, Khepera is made up of three layers corresponding to:

- A sensory-motor board equipped with eight infrared proximity sensors and two motors.
- A CPU board equipped with a Motorola 68331 microcontroller.
- A video board carrying a color CCD camera with 500 × 582 pixels.

Hardware

The host computer communicates with the robot via a RS232 link at the speed of 38,400 bits/s. The video signal is sent from the robot to the framegrabber in the PC, in a differential mode on two additional wires.

An external camera (Canon VC-C1) is mounted on the ceiling above the maze in order to give the user a global view. This is an interesting aid for planning a long displacement. A RS232 link between the PC and the camera allows the visitor to control the camera orientation and the zoom factor in order to have a better look at the current situation. The video is also wired to the framegrabber of the PC and it is multiplexed with the video signal coming from the robot camera. Therefore, it is possible to switch between the robot view and the external view. All these connections are summarized in Fig. 2.

Software

The PC operating system is Windows 95. The web server is the Personal Web Server from MicroSoft. This server launches several CGI (common gateway interface) scripts to perform the tasks. They communicate through shared named memory. Two programs run continuously:

- The first program grabs the images and puts them in shared memory in JPEG format (160 × 120).
- The second program puts the IP address of every new user in shared memory in such a way that other CGI scripts have the possibility to control the user identity. A timeout variable is regularly decremented in memory. After 5 minutes, if the user has not disconnected, this program puts the user IP address in a list forbidding him to obtain control again for 30 seconds.

When a client performs an access to the control page, the server starts a program that continuously sends the images stored in the memory. This image feedback is based on the server *push* technique supported by Netscape but not by Internet Explorer at the time of writing. Others techniques can be used but have not been tested in this experiment [6].

On the client site, the user has access to a control page created mainly in plain HTML and using clickable images. There are three types of commands available:

- Commands to control the robot movements (speed/position). The click coordinates are sent to the server, where a CGI script decodes and builds the corresponding order for the robot. The orders are sent to Khepera via the RS232 serial link.
- Commands to control the external camera movements (orientation/zoom). Here, too, the orders are sent to the camera via the RS232 serial link.
- Commands to switch the camera. A CGI script on the server acts on the multiplexer present at the input of the framegrabber.

A Java applet running on the client side regularly sends some requests for information about the state of the robot and the time left to the user. A CGI script on the server answers these requests by collecting the information from the robot and the shared memory.

There is no local "intelligence" on the robot such as obstacle avoidance. This type of mechanism is not necessary because of the light weight of the robot. This also means that there is no risk to destroy a wall or the robot itself. The advantage of having a direct control is that the user can see the result of his own action without any external contribution. The drawback is that control of the robot is more difficult without help and under important delays.

Interface

The client-level interface includes all possible operations that can be made on the robot, the external camera, and the multiplexer. The complete window available through Netscape is shown in Fig. 3.

The interface is composed of three columns corresponding to the three types of commands that the user can perform:

- In the left column there are the orientation controls (pan and tilt) as well as the zoom factor control for the external camera. The values of the parameters are given by clicking on the graduations.
- In the middle column there are the visual video feedback, the switch control of the two cameras, and the display of the robot status.
- In the right column, two different control panels allow orders to be sent to the robot. The upper one provides displacement commands. The lower one gives speed commands. Each of these control panels has two parts: one for the rotation and one for the straight movement of the robot.

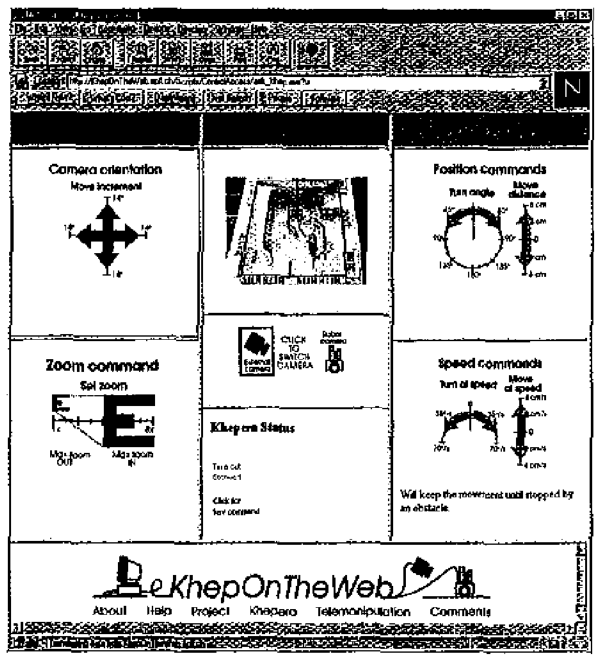

Figure 3. The remote control interface provided through Netscape.

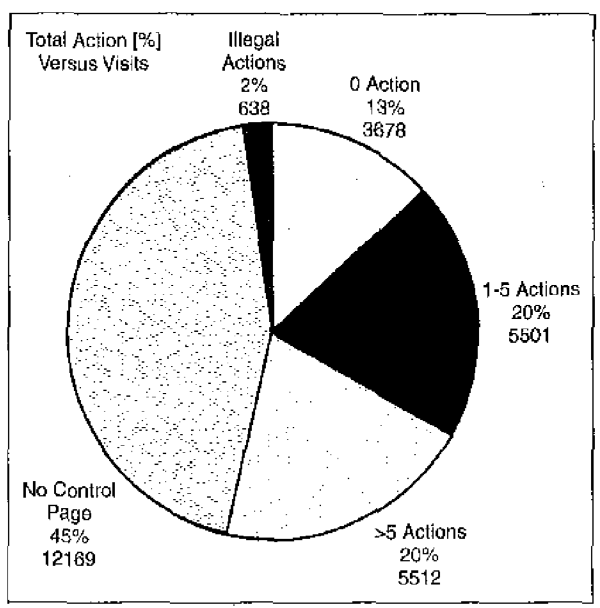

Figure 4. Distribution of actions versus visits.

This interface is not the first one we tried. In 1995 we started developing a control interface based on videoconferencing software (CuSeeMe). The results were interesting but the control software was running only on MacIntosh; we therefore moved to a pure web interface. The constraint is to send complete images to the client, but the performances are still good. It is also possible to use animated GIF files and to send only the part that changes in the images, but a "caterpillar effect" [6] can be observed. The image size is also small. However, for user comfort it is possible to display the image larger than what is shown in the browser without loss of bandwidth.

Finally, we have spent more time on the design of the interface than on all other engineering parts combined.

The visitor needs different view points in order to correctly understand the robot's location before deciding the next move.

Log Access Analysis

For this analysis, we consider the period between May 1997 and May 1998. As explained above, the access to the virtual setup is not taken into account. All our statistics are based on the IP addresses. Therefore, several persons from the same site will appear as a single user. Moreover, some Internet addresses are allocated dynamically, so the same person visiting on separate occasions may appear as another person. In this section, IP addresses, users, machines, and visitors have the same meaning. Next we define three terms: *action*, *session*, and *nip*.

Action is a script run by the user to control the robot or the camera. A *forbidden action* is an action launched by a user who has not asked for the control of the robot from the presentation page. He is not registered by the system and his action is refused. A *0 action* is due to a visitor who does nothing after having loaded the control page. If he did not load the control page, we do not count it as *0 action* but as *no control page*. A *session* or visit is defined as an uninterrupted access by the same machine with a maximal break of 10 minutes between two accesses. In *nip*, IP addresses in which the name could not be found in a DNS (domain name server) are gathered.

General Statistics

Based on these definitions, we had 27,498 visits performed by 18,408 unique machines. Only 3,178 machines did more than one access. Their average return time was about 23 days with a typical delay between two actions of 13.6 seconds.

The most active session was issued from the United Kingdom. The user performed 630 actions distributed as follows:

- 344 actions for the control of the robot.
- 243 actions for the control of the camera.
- 43 actions for the switching between the panoramic and the embedded cameras.

This particular visit had a duration of 1 hour and 10 minutes. This corresponds to an average of one action every 7 seconds. This visitor came back 13 times but he never did as many actions again.

Actions Distribution

All other sessions were not as active as the one discussed above. Figure 4 depicts the usage of the site. 45% of users visited only the welcome page at our site. They did not request control of the robot. There are two possible explanations for this:

- Browser type. Internet Explorer does not allow visual feedback using the server push technique. The welcome page contains a warning about this problem. Statistics in [11] show that the distribution between the two main browsers is 50% Netscape, 45% Internet Explorer, and 5% others.
- Type of access. The user has the option to control the real or the virtual robot. The welcome page of the virtual setup was loaded by 11,283 machines.

The percentage of users who had access to the control page but did not perform any actions was 20. The reasons for that could be:

- The robot was already being controlled by other people.
- The user could not understand the goal of the site.
- The image rate was too low (long delays) or zero (browser type).
- There were problems while loading the clickable maps or the Java applet.

Figure 4 represents very well the general actions distribution. We found approximately the same distribution in other analyses not illustrated here:

- Action distribution versus months.
- Action distribution versus number of visits.
- Action distribution versus domain of the machines.
- Action distribution versus our own use (demonstrations and verifications).

Figure 5 shows the effective use of the site, represented by the actions. The robot is mainly controlled in position. This emphasizes the "wait and

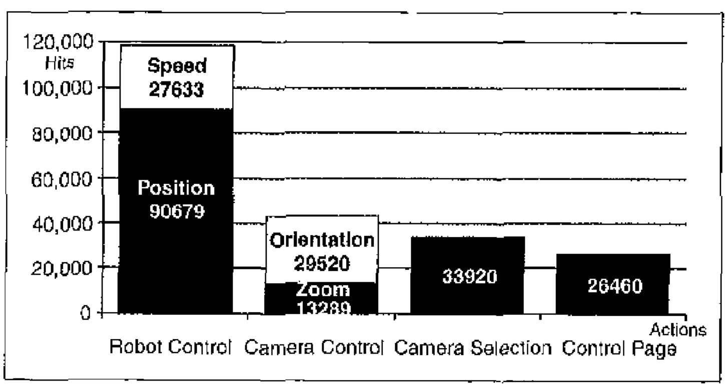

Figure 5. Actions to control the robot and the camera.

see" strategy developed by the users. Although the goal of the site is to control the robot, the camera is strongly solicited. Generally, a camera view is requested every second robot action. The visitor needs different view points in order to correctly understand the robot's location before deciding the next move. The importance of the column "camera selection" shows also the necessity to have both a panoramic camera and a camera on the robot.

The relationship between delays on the network and the number of actions performed is emphasized by Fig. 6. Only domains with more than 100 users who performed more than five actions are taken into account. There is a clear relation between the time/action and the number of actions/user.

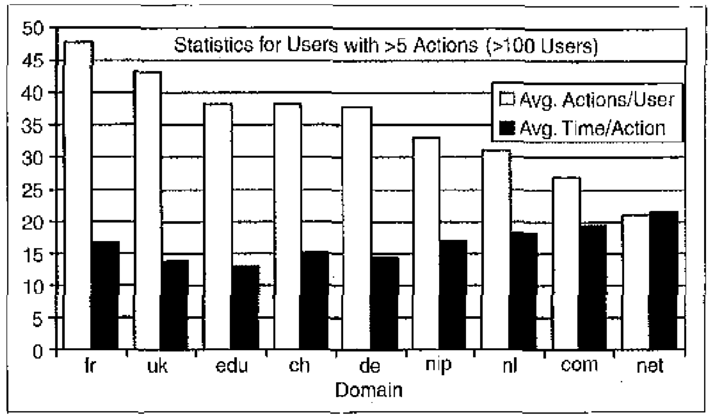

Figure 6. Relationship between delays and actions.

Figure 7 depicts the influence of external causes on the evolution of the number of accesses during the 12 months considered. The two months indicated by the arrows are interesting. August 1997 shows a minimum due to a holiday period [August 1998 (not illustrated here) shows similar results] and December shows a maximum. For the week of 8 December, the site was selected "Cool Robot of the Week" by NASA [12]. The number of accesses decreased the following month and stabilized around 2000 per month. Mainly, people visit once and do not return or come back soon a second and last time. We think that this behavior is based on the "surf effect" of the web. This result is verified by the statistics of the Australian Telerobot [13].

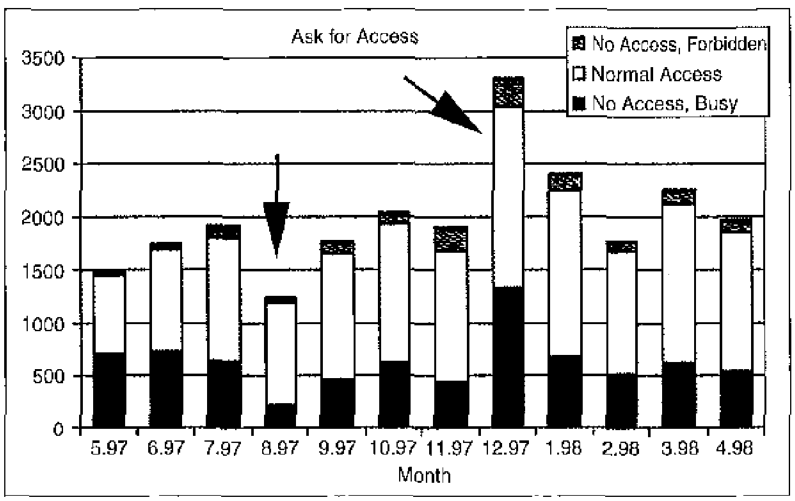

Figure 7. Accesses during 12 months.

Returning Visitors

Only 3178 machines came back to our site. They made 12,268 of the 27,498 visits. Different analyses about the returns are shown in Figs. 8 and 9. Although return average is about 23 days, Fig. 8 exhibits an important peak of returns centered on 18 minutes.

Of 9090 total returns, 1096 were made after 12 to 22 minutes. This confirms the "surf effect." This peak is mainly due to people who return to our site because they could not control the robot the first time (see Fig. 7), so they retry later. There is no return below 10 minutes because by definition two hits from the same machine made in an interval of 10 minutes belong to the same session

How do the returns influence user behavior? Figure 9 tries to answer this question. This figure describes the distribution of the actions for each visit for four user categories performing a different total number of visits. The analysis is restricted to the first four visits in order to maintain a representative sample. Users who come back have acted during their first visit much more than the ones who did only one visit. We could expect that from their second visit, users take advantage of their knowledge to do more actions. This is not the case. The number "of more than five actions" even decreases slightly. People who come back seem to only have a look at the presentation page. The reason could be that they want only to show the site to some other people, to get information, or to simply verify that the server still exists. The number of illegal actions increases as well. It is possible that users have left their browser open on the control page between the two visits. In this case, there is a time-out that makes the following action illegal. Most of the people come back after a short time (see Fig. 8).

Geographic Location Influence

All of the previous figures are established on the basis of the accesses of 18,408 unique machines. This is less than the 60,000 hosts that visited the Mercury Project site in six months [3]. One of their graphics shows that 58% of the machines were located in North America and only 14% in Europe. In our case, the situation is the opposite, with only 7% of the machines located in North America and 37% in Europe. This could be because the Internet was less well developed in Europe than in

North America. Another explanation is that our site runs differently. The Mercury Robot is controlled through the means of absolute coordinates and ours is through relative coordinates. It is therefore more difficult and time-consuming to guide our robot from one point to another.

Switzerland and its neighbors (France, Germany, United Kingdom) were responsible for most of the actions. The geographic proximity (reduced access time) favors the use of our site. In other words, our site is functionally less attractive for users far away because of the unacceptable response time. The domains *com*, *nip*, and *net* accounted for most accesses to our site, but they made fewer returns to it than Switzerland and its neighbors.

It is of great advantage scientifically to share a common environment for the understanding of the actual control approaches.

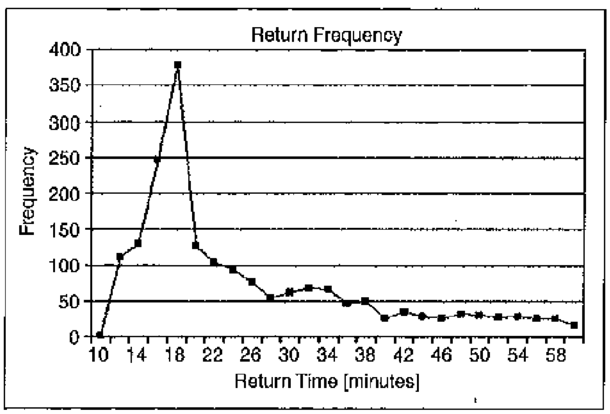

Figure 8. *Return frequency.*

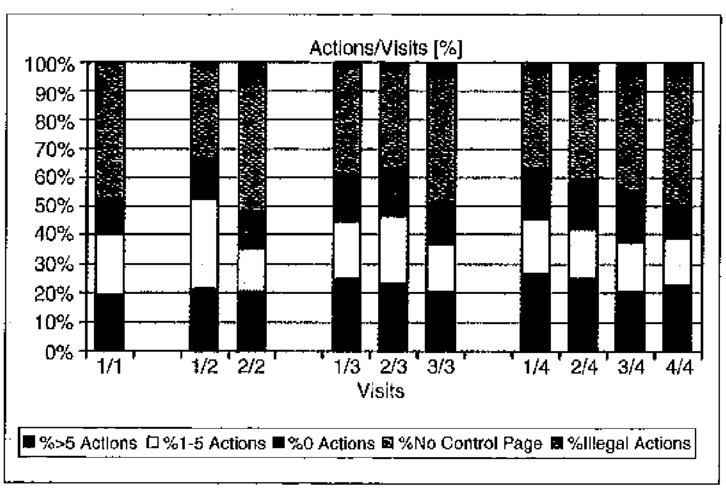

Figure 9. *User behavior versus number of visits.*

Future Work

The main goal of our project is both economic and scientific. On one hand, a mobile robotics setup is very expensive and could be shared between several research laboratories to reduce expenses. On the other side, comparisons between different approaches in the field of mobile robotics can only be made on robots placed in perfectly identical conditions. This does not mean that only the robot has to be the same but also that the environment has to be identical in every detail, such as lighting conditions or color and type of floor. It is therefore of great advantage scientifically to share a common environment for the understanding of the actual control approaches.

During 1998, another benchmark (TeleRoboLab) was built. It is reserved only for researchers and has been available since the end of 1998. It takes into account the results presented in this article but adds a scientific dimension to telerobotics. The new TeleRoboLab site provides the complete access to all functionalities of a much larger robot. The user is able to remotely control the robot or to download C code. The access has been restricted to registered scientific users. The setup complexity can be higher because the users are familiar with computers and robotics, but there is still a need for a good interface in order to let users concentrate on the algorithm without having to understand the site functionalities.

Currently, there is no tool available to cope with delays on the Internet [perhaps resource reservation protocol (RSVP) in the future?], but this network is widely accessible. To improve real-time access, the idea of our site is to be accessible from both the Internet and from another network (ISDN). ISDN has a poorer performance than other networks such as ATM, but it is less expensive and easier to install. The environment of the robot (lights, sliding door) can be controlled through a web interface running Java applets.

At the time of this writing, the concept presented in this section has only been validated with local experiments.

Conclusions

The experience accumulated in this first test shows some important aspects. Such a physical environment is technically feasible with commercial parts. Everything from the robot to the camera and from the server to the framegrabber are very standard commercial products. A complete Web robotics package based on our experience is available on the market [15]. But there are still conceptual problems. One is to introduce a complex setup on the web where the rule is more "click and wait" rather than "read and click." Another problem is obviously the delay that prevents people from having a good interaction and from taking interest in the site. This is not the only reason why users do not come back. Another reason is that the site is frozen: there is nothing new to see in a second visit; once you have controlled the robot, there is no reason to come back. As said above, you have to catch the netsurfer's attention the first time he comes. It is interesting to observe that the behavior of the users is generally independent of the country or the time the site is available, which shows that there is a global and stable Internet culture.

However, the analysis of Internet user behavior is not simple. The graphics and the analysis of the previous paragraphs show well that a good understanding is difficult to obtain. Only a few of our analyses are included in this article. Significant graphics are very difficult to calculate because of the huge quantity of data of the log-files. Even with such an amount of data, there were still categories with not enough representative samples. When a graphic exhibits a feature, it is rather difficult to explain it with the general knowledge about the Internet. Moreover, you have very little feedback from the users themselves. We received only one piece of mail for 1000 accesses, and it contained very little information.

The reliability of the setup is good but we had to face two main problems. As said above, the PC server is running Windows 95. This system is not stable enough to allow the machine to operate without being regularly reset to prevent a crash. The Personal Web Server was not well adapted for our usage. Sometimes it froze in a given state, and it was then impossible to access the welcome page. This kind of test has been useful for a new project called RobOnTheWeb [14]. The project team has an additional reason to use Linux as operating system and Apache as the server.

The other problem is rather mechanical. Sometimes the robot is prevented from moving by an edge of a wall. Khepera is very light and its two motors are powerful. Therefore, against a wall, it is possible that it rises a little and that one of its wheels loses contact with the floor. On the ramp, wheels can have less adhesion due to the weight of the cables or due to dust, also making it difficult to control the robot.

Acknowledgments

We would like to thank Edo Franzi and Andre Guignard for their important work in the design of Khepera. We also thank Prof. Jean-Daniel Nicoud and all the LAMI assistants who have tested our interfaces for their useful comments.

Keywords

Mobile robots, Internet, web-based control, remote control, user behavior

References

[1] R. Simmons, "Where in the world is Xavier, the robot?" *Rob. Mach. Perception*, vol. 5, no. 1, pp. 5-9, 1996.

[2] K.A. Golberg, "Telerobotic garden on the World Wide Web," *Rob. Mach. Perception*, vol. 5, no. 1, no. 4, 1996.

[3] K.A. Goldberg, "Desktop teleoperation via the World Wide Web," in *Proc. IEEE Int. Conf. Robotics and Automation*, p. 654-658, 1995.

[4] J.A. Fryer, "Remote-control experiment using a networked robot." *Rob. Mach. Perception*, vol. 5, no. 1, p. 12, 1996.

[5] O. Michel, P. Saucy, F. Mondada, "KhepOnTheWeb: An experimental demonstrator in telerobotic and virtual reality," in *Proc. Int. Conf. Virtual Systems and Multimedia* (VSMM'97), IEEE Computer Society Press, 1997.

[6] http://rr-vs.informatik.uni-ulm.de/rr/

[7] http://graco.unb.br/robwebcam.html

[8] http://freedom.artm.ulst.ac.uk/~antonh/research/Lynxmotion.html

[9] http://www.yahoo.com/Computers_and_Internet/Internet/Entertainment/Interesting_Devices_Connected_to_the_Net/Robots/

[10] http://queue.ieor.berkeley.edu/~goldberg/art/telerobotics-links.html

[11] http://www.cen.uiuc.edu/bstats/latest.html

[12] http://ranier.oact.hq.nasa.gov/telerobotics_page/coolrobots97.html

[13] http://telerobot.mech.uwa.edu.au/

[14] http://diwww.epfl.ch/lami/team/michel/RobOnWeb/

[15] http://www.k-team.com

Patrick Saucy received his master's in electrical engineering from the Swiss Federal Institute of Technology of Lausanne (EPFL) in 1996. Since May 1996, he has been working as a research assistant and Ph.D. student at LAMI-EPFL. His Ph.D. focuses on distributed environments to control remote robots. His research interests include hardware tool design to enhanced remote experimentation and software tools to evaluate supervision versus direct control.

Francesco Mondada is a microtechnics engineer from the EPFL. He received his master's in computer science from the EPFL in 1993 and his Ph.D., also from the EPFL, in 1997. Since 1991, he has been working on mobile robotics, especially on the miniature mobile robots Khepera and Koala, both developed with Edo Franzi and André Guignard. Co-founder of K-Team SA, he splits his time working at this company and working at LAMI. His main interest is the development of methodologies for the design of control structures for autonomous mobile robots.

Address for Correspondence: Patrick Saucy, Microprocessor and Interface Lab (LAMI), Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland. E-mail: patrick.saucy@epfl.ch. Web: http://diwww.epfl.ch/lami/team/saucy