# A Multimedia On-line Assistance System with Safe Remote Control of Applications*

P. Sylvestre, M. Barbeau, and R. St-Denis
Département de mathématiques et d'informatique
Université de Sherbrooke
Sherbrooke, Québec, CANADA J1K 2R1
{barbeau, stdenis}@dmi.usherb.ca

## Abstract

*This paper tackles the problem of efficiently helping software end users. Assistance to end users has become increasingly critical to the productivity of organizations. The cost of a person or a group of people that provides assistance is substantial. Besides, safety and confidentiality of end-user information are often compromised when assistance is provided. Remote assistance using remote-control and visualization software is one of the means to efficiently provide assistance to end users. Several software systems have been proposed to take the control of remote computers and applications, but few of them take into account human aspects and guarantee security and confidentiality of information. Consideration of these elements is necessary for practical remote resolution of problems.*

*Using concepts from the field of groupware and remote control of computers, we have devised software to provide remote assistance to end users while insuring security and confidentiality of information as well as providing the tools required for natural human interaction. This paper presents our software.* *keywords:* on-line assistance, remote control, end-user assistance, distributed multimedia systems, groupware.

## 1 Introduction

A recent study by the Gartner Group [3] brought to the fore that the total cost of ownership of a PC type workstation is between US 8 000*toUS* 13 000 per year. Cost of ownership can be attributed to four types of expenditures: the actual cost of the workstation, end-user support, administration, and end-user operations (e.g., installing new software, excluding strictly job-related tasks). These expenditures account respectively for 21%, 27%, 9%, and 43% of the total cost. Hence, there are potentially big savings that can be made in the areas of end-user support and operations. This paper presents software that can improve the way things are done in these two areas, thereby reducing the total cost of workstation ownership.

Most organizations have a consultant or a technical-assistance department to help end users to limit their down time. Providing technical support certainly represents an important share of an organization's budget. Requests for technical assistance have to be served efficiently. The technical assistance staff is often forced to travel to end-user sites, which fritters away precious time. In addition, problem solving may necessitate calls to expensive external resources.

There are four types of difficulties encountered by end users: hardware, hardware configuration, software, and software configuration. Hardware difficulties occur when a piece of equipment fails. For example, a hard disk crashes or a screen blacks out. In these cases, the failed hardware has to be replaced or repaired. Hardware configuration difficulties occur when installing new equipment. Parameters are inadequate or they conflict with parameters of existing equipment. For example, a new processor requires adjustment of the mother board frequency. Software difficulties occur when an user is unable to use software functionalities. For instance, the procedure for printing or saving files. In this case, the user could be walked through the steps. Software configuration difficulties occur when installing new software. They are caused by inadequate parameters and conflicts with existing software. An improper entry in the *autoexec.bat* file is an example. The purpose of the research presented in this paper is to develop solutions to improve the quality of assistance and to reduce time spent by end users and consultants on solving difficulties.

We have created software that establishes a dialog, through a telecommunications network, between the

*The research described in this paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

workstation of an end user, in trouble, and the consultant's. Our software allows the consultant to remotely visualize and control the application running on the end-user workstation. The consultant, therefore, can rapidly troubleshoot the problem and solve it without physically moving. In addition, our software uses a videoconferencing technology to make interactions between parties as efficient and as natural as possible.

A substantial number of difficulties can be handled without consultant travel, while reducing costs. Only software and software configuration difficulties, however, can be easily solved with our approach. Hardware and hardware configuration difficulties lend themselves less to resolution by remote observation and control. The reason is simple: hardware failures impede or prevent th system from functioning.

There are commercial software packages offering facilities similar to those that we developed, including PC Anywhere 32 [www.symantec.com], Laplink [www.travsoft.com], NetOp Remote Control [www.danware.com], ReachOut [www.stac.com], and Remotely Possible [www.avalan.com]. They allow remote visualization and application control. They often suffer from flaws, however; security is among the major concerns because the consultant can see and modify confidential information on the end user's system. There are two possible alternatives: hide the confidential information from the consultant or don't ask for help. These software systems offer no control over information access. Furthermore, support requests often require complex manipulations on the part of the user. For example, the user has to open assistance software and provide parameters to establish a logical communications link between the assistance system and its application. Our software includes mechanisms for protecting confidential information and simplifies making assistance requests.

In our system, the consultant sees only the window of the shared application. Each piece of information and every operation in the application can be protected independently of the others. This provides a high level of security and privacy. Since the consultant has only partial access to an application, some configuration problems, however, cannot be solved. The consultant, nevertheless, is able to diagnose configuration problems if they affect the application.

Both the consultant and end user have the same view of the application and can interact with it using a shared cursor or a shared pointer. Input from both participants is reflected by their cursors and pointers. The consultant gives explanations in the end-use environment. The end user obtains instructions directly on his workstation. In addition, our software instructs an agent to automate the formulation of assistance requests and to protect against access to confidential information. It reduces the amount of skill required by the end user.

To implement our software, we opted for the Microsoft Windows 95/NT operating system, mainly because it is one of the most widespread. Moreover, we have opted for the Borland Delphi [www.borland.com] object-oriented programming environment.

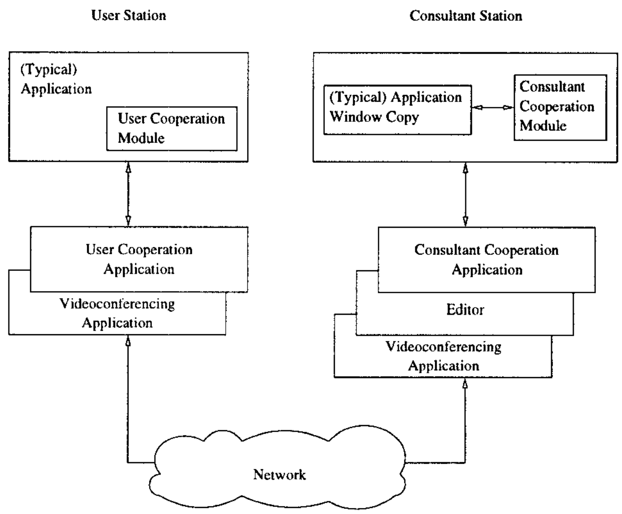


Figure 1: Architecture of the software

The architecture of our software is illustrated in Figure 1. It comprises the end user and consultant parts. The end user (consultant) part comprises all modules and applications that run on the end user (consultant) workstation. Our software only shares applications that are specifically designed and programmed for our environment. We developed a typical application for demonstrating the realism of our concept. The user cooperation module is transparently embedded in the application. The end user notes only the presence of an extra assistance icon. Clicking on this icon triggers an assistance session handled by the user cooperation application. The user cooperation application is present to provide remote cooperation facilities to the end user workstation and to the consultant. This application partly provides security, text mode communication, and the launch and termination of a videoconferencing application. The videoconferencing part is handled by commercial products from Connectix [www.connectix.com].

The consultant has on his workstation an exact copy of the typical application window, on which he can act. Consistency of the copy with the original window is maintained in real time during a session by the consultant cooperation module. The consultant cooperation application provides cooperation facilities: text mode communication, remote file editing, and activation of a videoconferencing application. The consultant can use a text editor to access some configuration files stored on the end-user workstation. A list of files is provided to text editor at the beginning of the session and only those can be read and modified. A commercial videoconferencing product is also integrated.

An assistance session typically goes as follows. Firstly, a user triggers the process by clicking on the assistance icon. An agent is then activated to help the user proceed. The agent prompts the user to select which information to hide and which operations to disable. Then, the agent starts the user cooperation application, which is required to establish a session with a consultant. Messages are transported over the network. The cooperation applications transmit and receive them. Upon request, they activate the videoconferencing applications. The consultant cooperation application activates the copy of the typical application window. The assistance session is now under way. Either the end user or the consultant can terminate the session.

The rest of this paper is structured as follows. Section 2 presents an overview of the user interface of our assistance software. Section 3 reviews some of the key mechanisms in the design of our software. We conclude with Section 4.

## 2 The Assistance Software

This section presents a more detailed look at our assistance software. We have developed a typical application in order to demonstrate the functionalities of our software. The typical application manages records, each of which is composed of personal data. The application user can add, modify, suppress, and browse through records. Records are stored in a database (Borland Paradox). Figure 2 pictures the main window of this application.

Our assistance software makes possible cooperation between two participants that share a software application over a network. Participants have asymmetric roles: one of them acts as the end user; the other as the consultant. The software implements mechanisms that keep each participant aware of activities performed by the other. Moreover, these mechanisms insure protection of confidential information. A significant strength of our software is its integration into

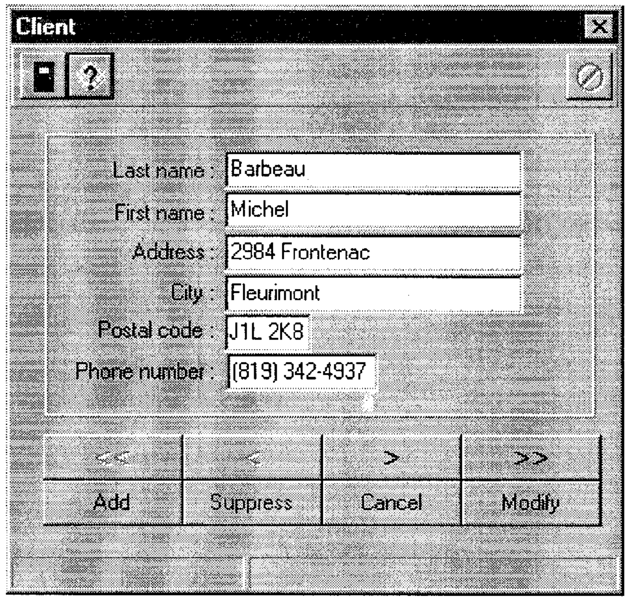


Figure 2: Main window of the typical application

the client application, for simpler use.

The specifics of our assistance software are further described hereafter using a typical session. The session begins with an assistance request generated by the end user (by clicking on the question mark icon; see Figure 2).

An agent is started and takes charge of a procedure for handling the assistance request. At the beginning, and before connection with the external consultant, the agent offers the user the possibility of hiding data fields and disabling operations, for privacy and security purposes. By default, some fields are hidden and some operations disabled. Windows of selectable fields and operations are presented to the user (see Figures 3 and 4). Then, the connection is established with the consultant. A cooperation application and a videoconferencing application are started.

The assistance session is now in progress (see Figure 5). Whenever one of the participants uses the mouse or the keyboard, the action is automatically reflected on the other's screen. The end user and the consultant can converse by videoconferencing or in text mode via the cooperation application.

From the consultant's point of view, an assistance session flows as follows. A cooperation application is started on the consultant's station and waits for an assistance request. When a request arrives, a copy of the client window and a videoconferencing application are

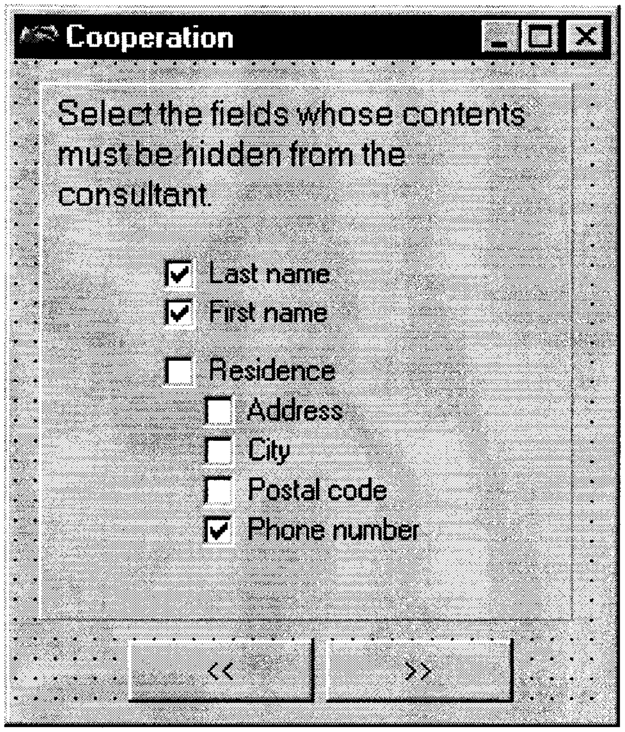


Figure 3: Hiding data elements

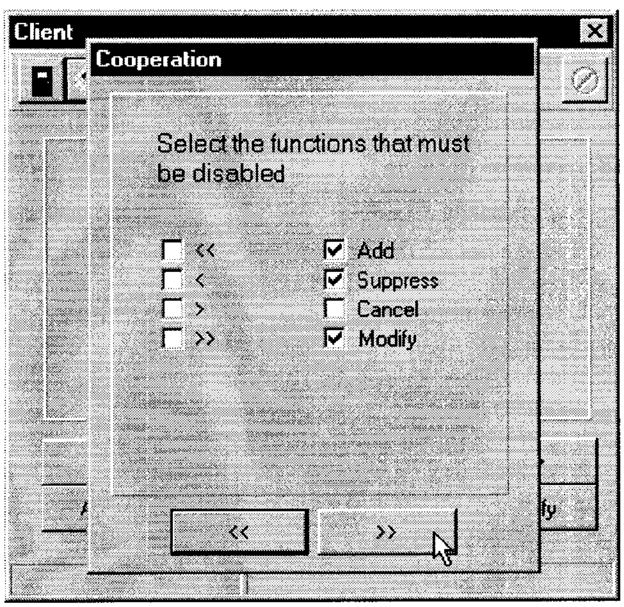


Figure 4: Disabling operations

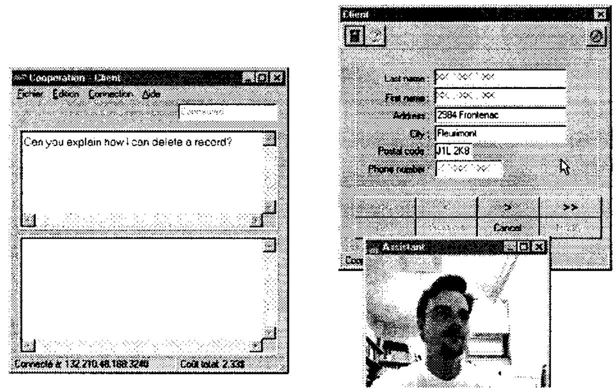


Figure 5: Assistance session

activated. The consultant can then receive questions from the user. The consultant can use a remote text editor to inspect and modify files on the user workstation. A list of accessible files is provided to the consultant station by the end user.

## 3 Key Mechanisms

In this section, we review some of the key mechanisms employed in the architecture of our software. The design uses an object-oriented approach [1].

### Communication Mechanism

Cooperation between the different components of our software necessitates network and interprocess communication. For network communication, the TCP/IP protocol suite has been adopted and is accessed through Microsoft Winsock [4]. TCP/IP was selected because of its availability on personal workstations. Technically, however, it is certainly not the best choice because there are protocols better adapted to real-time communication that can meet real-time restrictions. In practical terms, however, they are not as widely spread as TCP/IP.

Two types of interprocess communication are used: Dynamic Data Exchange (DDE) and messages [2] [5] [6]. DDE is used to transfer the contents of fields which are often too large to be transferred by messages. Message-oriented communication is used to exchange information about actions (e.g., mouse movements) that need to be transferred in real time.

Time constraints have to be satisfied in order to make the assistance process as efficient as possible. The performance level of a DDE connection is easily affected by variations in workstation load and power. On the other hand, message-based communication encounters practically no performance variations and easily meets our time constraints. Each message, however, can transport a limited quantity of information (two long integers). In contrast, a DDE connection can transport a large quantity of information.

### Remote-control Mechanism

During an assistance session, the end user shares the control of his application with a consultant. The latter has a copy of the window of the end-user's application, which is automatically updated whenever a modification occurs. Actions of a participant (mouse movement and keyboard input) are reported to the other participant and reflected in his window. Only one pointer and one cursor are logically present; both participants can modify their positions. A participant may, for example, observe a mouse movement without performing any action. Synchronization of the mice generates bursts of small packets (containing $x$ and $y$ coordinates) over the network.

### Security Mechanism

Implementing software that includes remote-control mechanisms requires information security procedures. In our software, only the user part requires security mechanisms because only its information is shared. An end user can connect to a consultant's workstation, but the reverse is not possible. Therefore, the end user is aware of the fact that his application is in part accessible to another party. The end user environment-related information is protected by the fact that his application window is shared. Information contained in a shared application is protected by a nondisclosure strategy. Before the session begins, the user is requested to select data fields and operations that will be inaccessible to the consultant. Afterward, the software replaces the real data by dummy symbols and puts the selected operations out of order. During the session, neither the participants will be able to access the real information or to use the disabled operations. Of course, the security of information that travels through the network must be insured. This may be provided with of-the-shelf encryption technology, although it has not yet been integrated into our software.

### Awareness Mechanism

Awareness mechanisms are introduced to make the explanation session as productive as possible. The videoconferencing link makes each participant aware of the other's presence in the environment. The shared pointer and cursor make both participants aware of actions performed by the other.

### Remote Edition Mechanism

A remote editing mechanism allows the consultant to visualize and modify the contents of files stored on the end-user workstation. The consultant part has an editing window with the usual editing commands. An editing module residing in the end-user part controls access and actions to files. Moreover, it reports and apprises the end user of the consultant's editing.

### Videoconferencing Mechanism

Videoconferencing communication relies on the use of commercial software. Any generic videoconferencing software can be used for this purpose as long as it can be started with an IP address in parameter and stopped from another program. This allows the client to start the videoconferencing software with the network address of the consultant workstation. We selected VideoPhone of Connectix [www.connextix.com] because it provides a reasonable quality of service with good performance.

Videoconferencing is the major source of network traffic. In addition, compression and decompression of video signals is the most demanding CPU time task of our software. Bandwidth availability and relatively powerful workstation are required for our software.

subsection*Agent Mechanism

Since our software targets novice end users, use me be kept as simple as possible. The end user can request assistance simply by clicking an icon and a wizard takes care of the details. The assistance wizard helps the user by automating the connection and configuration processes of the cooperation environment.

## 4 Conclusion

With a growing number of end users, computers have become an integral part of an control production, validate processes, and perform various administrative tasks. Training of employees that use computers every day is crucial to insure a high level of productivity. Good training, however, is not sufficient to achieve and maintain high productivity. Technical support staff are needed to solve problems that are beyond the field of expertise of end users. Several computer technologies presently exist in order to maximize the productivity of technical staff. Some of these tools provide on-line assistance.

The contribution brought by our research is software implementing a series of principles that improve the way technical on-line assistance is provided to end users. Our software insures security and confidentiality, and is easy to use because it is integrated into the end-user application. We proposed several rules in order to improve security as well as confidentiality while reducing the amount of information shared by participants. Our software, however, has some limitations. One of these limitations is its prototype character. It concretely demonstrates and illustrates principles, but it is not yet a marketable product. A more robust and efficient version of our software would allow its use on a large scale.

The number of end users of computer technologies in organizations and homes brings new challenges to hardware designers and, more particularly, software. The quantity of novice end users is high and their need of assistance grows with the arrival of new technologies. The area of end-user assistance therefore will become an increasingly predominant subject for organizations, manufacturers, and designers of computer solutions.

## References


[1] G. Booch, *Object-Oriented Analysis and Design with Applications,* Second Edition, Benjamin/Cummings, 1994.

[2] C. Calvert, *Delphi2 Unleashed,* Borland Press, SAMS Publishing 1996.

[3] R. Comerford, "The battle for the desktop," *IEEE Spectrum,* May 1997: 21–28.

[4] B. Quinn and D. Shute, *Windows Sockets Network Programming,* Addison-Wesley Advanced Windows Series, Addison-Wesley 1995.

[5] R. Simon, M. Gouker, and B. Barnes, *Windows 95 Win32 Programming API Bible,* The Wait Group 1995.

[6] R. Simon, *Windows 95 Common Controls and Messages API Bible,* The Wait Group 1996.