VSSa94a] *Marcus Wieland, Ralf Steinmetz, Peter Sander;* **Remote Camera Control in a Distributed Multimedia System**; Innovationen bei Rechen- und Kommunikationssystemen (24. GI-Jahrestagung), Hamburg, B.Wolfinger (Hrsg.), Springer-Verlag, S.174-181, 1994.

Remote Camera Control in a Distributed Multimedia System

Marcus Wieland, Ralf Steinmetz, Peter Sander

IBM European Networking Center
Vangerowstraße 18 • 69115 Heidelberg • Germany

1 Introduction

Most publications on multimedia issues address basic technology or system software aspects (video server implementation, communication protocols, etc.). Some papers discuss networked applications with the focus on multimedia conferencing, multimedia and hypermedia documents and video on demand with client and server systems. We look into details of networked applications, a remote camera control application.

We designed and implemented the HeiCam application (HeiCam - The Heidelberg Remote Camera Control) to exploit the advantages of existing computing and communication for the purpose of a remote camera control. Remote camera control facilities are mainly used to observe buildings or restricted areas, especially where danger to life exists or employment is unprofitable, as in chemical plants or nuclear power stations. First, let us describe in detail how a remote camera control works in a networked environment.

The camera is attached to a computer, the "camera server", via an electrical interface, for example a serial RS232C interface. Control commands like "move", "zoom" and "focus" are sent via this interface from the computer to the camera. The originator of the commands is an application running on a remote computer, the "camera client". Both computers, client and server, are interconnected by a computer network, for example a Token Ring or Ethernet LAN. Audio and video data is continuously captured, coded and compressed at the server. This data is transmitted over the network to the client. At this remote location the audio/video signal is displayed in the window of the client application. In these set-ups the camera attached to the camera server can be controlled remotely from one of the clients. The user controls horizontal and vertical movements by clicking with the mouse on appropriate buttons or on some part of the displayed video image itself. All other control functions of the camera are also reflected appropriately at the user interface at the client workstation.

We learned that the visual control of critical production processes would be easier if video pictures from the process could be displayed together with corresponding measurement data on a single display. Control consoles for production supervision have always been developed with the goal of integrated solutions: Control personnel should have all relevant information at their disposal in a compact and precise form. Up to now it has not been possible to integrate video using the same facilities. Consequently separate displays had to be installed in control centers. There are advantages to retrofitting video monitoring into existing control consoles. Video pictures can be shown in one window while other windows display measurements and alarm indicators along with schematics of the production process. With an intelligent presentation system, it is possible to display only relevant video information at a certain point of time or at a specific state of the industrial process to be supervised. The physical distance between camera and the controlling workstation is of no concern. The control commands for the camera as well as the live video and audio data are transmitted over the same digital network. Other scenarios based on the same live video technology include the use of cameras in security systems.

2 Environment and Constraints

HeiCam has been developed in various versions over the last 3 years. Based on the initial prototypes a user requirement study was performed. We wanted to find out the specific user demands of what and how to remotely control a camera. In the following we outline these issues and describe how they are reflected in HeiCam. We have developed HeiCam on PS/2 with OS/2 [4,3] and on RS/6000 with AIX [9].

Network Environment

A main motivation for the use of this application is the desired reduction of expenses in association with the installation of any distributed remote camera control. Existing infrastructure, namely computer networks and the attached workstations should be used. It should be avoided to invest in dedicated equipment such as analog cabling for the transmission of video signals.

During the last years, there is a discernible trend from stand-alone workstations and PCs to networked systems. A networked environment offers the opportunity to share expensive peripheral devices such as laser printers, scanners and in principle also audiovisual equipment like cameras. The other principal usage is the exchange of data. Conventional LANs provide fast bulk data transfer in a fair mode without real-time demands. To a certain extent these networks can also function as "shared medium" for the exchange of audio/video data.

In such an environment a workstation can act as client and/or server. HeiCam incorporates this client/server paradigm: Several servers and several clients are attached to the same network. The server provides video data to the clients using multicast facilities of the communication protocols. Any client is able to control every camera attached to any server. Servers have to be equipped with special video hardware for framegrabbing, digitizing and compressing the desired video signal. Depending on the employed compression scheme the client might also be equipped with hardware to speed-up the decompression and display process.

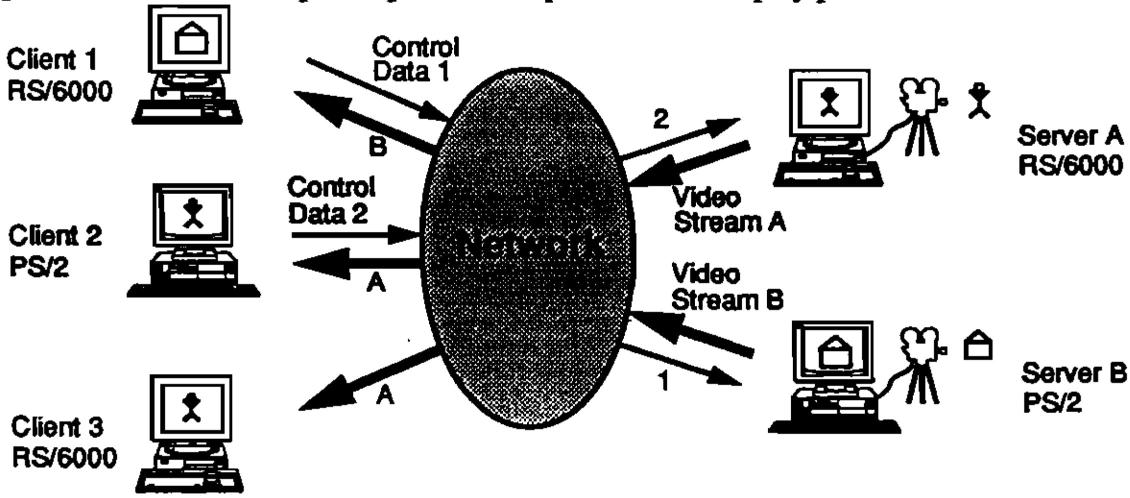

Figure 1: Network Environment

Figure 1 shows a typical scenario of the HeiCam application and symbolizes the network environment: Server A and Server B feed their video data into the network. The clients can choose which video streams they want to attach to. Client 1 "consumes" the "video stream A" and controls the respective camera with the "control data flow 1". Client 2 displays the video data of stream B and sends the control data flow 2. The HeiCam application at client 3 can display either video stream.

Independence of a Specific Audio and Video Format

As outlined in [7,8] there already exist many compression techniques. The most important compression techniques are JPEG, H.261, MPEG and proprietary developments like DVI, Intel's Indeo, Microsoft's Video for Windows, IBM's Ultimotion, Apple's QuickTime or DigiCipher II developed by General Instruments Corp. and AT&T. This large number of compression schemes and their ongoing developments demands for adaptability to various formats and implementations. The use of a seperate AV subsystem allows the integration of further compression techniques and capture/display hardware adapters outside the scope of the Hei-

Cam application. Therefore the HeiCam application is independent of any specific audio and video format.

Independence of a Specific Camera Type

Similar to the various formats of audio and video data and their integration into the audio/video subsystem, a closer look to the diverse camera hardware control interfaces is required.

The architecture of a distributed remote camera control should support a large amount of video cameras. These cameras differ considerably with respect to their intended use (for example monitoring or teleconferencing). We encounter considerable differences in the hardware interfaces for each remote camera control. HeiCam must allow for the integration of this set of heterogeneous hardware resources. Hence in analogy to well-known operating system components we introduce a novel interface software between control functions and the actual camera hardware: the "camera driver". A set of drivers for different cameras provides the independence of a specific camera type.

Functional Blocks

A structured design of the remote camera control application results in the separation of the software into the following functional blocks.

- User interface (text and graphics oriented)
- Networked control functions for the camera control (including a camera driver at the server site) with the "HeiCam Control Protocol" (HCP)
- A networked audio/video subsystem with the "Source and Sink Control Protocol" (SSCP)

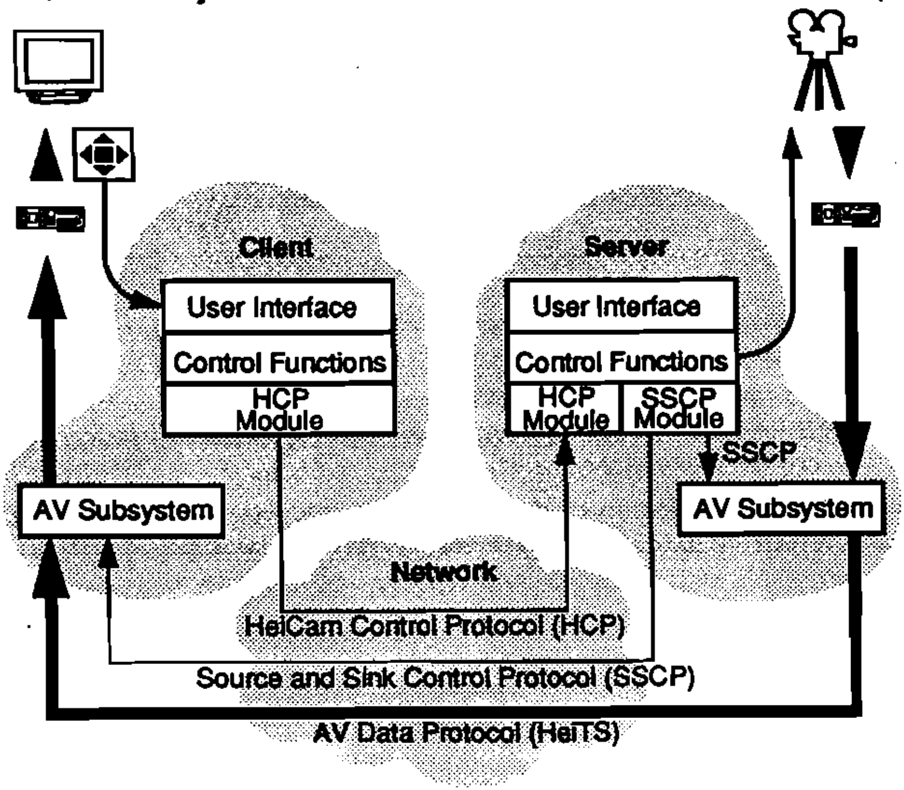

Figure 2: Data Stream and Control Stream Flows

The user interface enables the user to interact with the HeiCam application. It must provide a clear view of the available functions for the control of the camera which must be easy to use and easy to remember. The control functions module performs the commands issued by the user via the user interface. At the server site it processes the communication with the camera driver. Protocols for the communication between client and server and the communication between server software and the separated audio/video (AV) subsystem are needed. Therefore, interface modules to these protocols are required. The audio/video subsystem controls and manages the distribution of the audio and video data from a server to the several clients. It is developed separately, interfaced by HeiCam and entirely encapsulated. Note that also in our version with an object-oriented design and implementation we encounter the same functional partitioning as part of the class hierarchy [3].

Video Stream and Control Protocols

As mentioned in the description of the functional blocks above, two control protocols are needed to negotiate and establish the actual video data transfer, and to control the remote camera itself. These protocols exist in addition to the protocol for the transmission of the video images. The protocols and the video stream are all transmitted over the digital network, however they have different requirements:

The transmission of audio and video data has to fulfil real-time requirements and must obey strict time constraints. A management of the involved resources, e.g., network bandwidth, CPU time and buffer space, has to be done. Therefore, a special transport system for audio and video data is needed. The Heidelberg Transport System (HeiTS) [2] processes the transmission of the AV data and ensures the needed resource management. In principle any other multimedia transport system can also be used as long as it interworks with the available AV subsystem.

The communication between the server and the AV subsystems of server and client requires a protocol to cause the AV subsystem to start or stop an audio/video transmission between server and client. This protocol is not strictly time-bound and it should provide for the selection of different video data rates, constant and variable bit rates as delivered by the respective compression schemes. It is known as SSCP, the Source and Sink Control Protocol [1].

The stream of camera control data is not strictly time-bound, but it should be reliable. This control protocol is called HCP (HeiCam Control Protocol) and is based on RPC. The use of traditional protocols, such as TCP/IP, as transport layer protocols of the RPC are suitable for the transport of the control commands and simplifies the exchange of these commands in heterogeneous environments.

3 Specific User Requirements

Besides the consideration of the above elaborated application environment and the introduction to the basic functional blocks, the most crucial factor of success in any multimedia application is a clear understanding of the specific user requirements.

Any user interacting with a remote control application is not only interested in having as many features and functions as possible; she/he demands for a fast response time to any given command and it should be very easy to use for a casual as well as a professional user. All these major points are the primary factor for the success or failure of HeiCam. All architecture and implementation issues must support and enable these user requirements.

Control Operations

In the design phase of a remote camera control, the question arises which control operations should be available for the user and how they are called. First of all, the user expects an intuitive way of handling the remote camera: Selecting and changing the shot the camera offers. The user needs, for better acceptance, hands-on movements and focus and other operations of the camera, as he would hold a camera right in his hands.

The HeiCam user interface offers therefore a set of functions a user will expect. There are

- Move (left, right, up, down), zoom and focus
- Video capture (with facilities to record and playback videos at the local machine as well as at a remote server. This can be used if a security guard or supervisor is recognizing an unauthorized access to a restricted area and the video is serving as evidence.)
- Defined camera positions (storage and retrieval of several camera positions in order to

access predefined camera positions rapidly. These positions are defined by the spatial position of the camera and the corresponding zoom and focus values.)

- Adjustments for image quality (like contrast, colors, brilliancy, etc.)

Access Control

The acceptance of any video camera in a remote environment highly depends on the whole access control support including authorization. The system architecture should support a variety of possibilities as enumerated in this section.

With respect to the available capabilities, the access to a remote camera control can be allowed for all operations, for a subset of operations, for no control operations of the camera but display of video data or for no operation at all.

With respect to the users, machines and applications, the access to a remote camera control can be allowed to a person or to a group of people, to certain applications or to computers or groups of computers (i.e. for anybody working in on machines owned by a department).

If several people/applications/computers are authorized simultaneously, the access to the camera has to be synchronized. Just imagine a soccer game transmission where each spectator can control the same camera. Therefore, the possibility to control the camera has to be granted exclusively to a person/application/computer at one point of time.

User Interface

Many multimedia applications in the domain of conferencing are technology driven. Hence the user interface has not the highest priority. We see this from a different perspective, and are experienced that the users are most critical concerning this issue: The success or failure of an application like HeiCam heavily depends on the acceptance of the user interface by the user.

The user sitting in front of this computer wants to change the position of the camera in order to get another view. Therefore, he issues control commands to, for example, move, zoom, and focus the camera. He interacts with the application through the user interface of the camera control application. The goals of the design of such a user interface are to provide ease of use for first-time users, an easy to remember mechanism for casual users, and a fast and effective operation technique for frequent users. To some extent these goals are contradictory and difficult to implement. We approached this crucial issue by providing a button/menu based window interface as well as the novel "active video window" approach (as outlined in the next section).

In the novel approach the movement comprising "up", "down", "right", "left" i.e. rotations with respect to two right-angled axes of the camera are controlled by immediate interaction with the moving image [5].

Fast Response Time

Another requirement following the interaction between user and user interface of the multimedia system is the already mentioned real-time and a short response time.

The purpose of a remote camera control system in a multimedia environment has the possibility of controlling the "live action". It is especially the interaction between user and system that ensures a high acceptance of the overall system. This way, the user gets an immediate response of his operation shortly selected before and a precise adjustment of the camera, i.e. for focus operations. A multimedia system that is unable to meet these assumptions implies unwanted delays and a long response time.

4 Implementation Aspects

To separate the application into functional blocks, a special block of the control functions was built. The control functions are the logical part of the performance of control commands given by the user on both sites, client and server. Because of the independence of a specific camera hardware the control functions are divided into device independent control, the logical part of the control functions, and a device dependent control. Both parts are connected with a clearly defined programming interface. The part of the device dependent control is called the camera driver and performs the transmission of control codes to the camera hardware.

This existing architecture and implementation of a remote camera control fulfils all the requirements mentioned above. This realization of the requirements is described in this section as well as the used communication protocols. This section shows how HeiCam works and lightens the most critical aspects.

Camera Movements

The camera movements contain all functions for horizontal and vertical move, zoom and focus of the camera and its lens. These movements can be combined because of a similar performance of their commands. The HeiCam Control Protocol (HCP) was designed and implemented for the transmission of the control commands from client to server.

There are two possibilities to formulate a control command for a camera movement: A relative command or an absolute command. A relative command is a control command like "Move camera 180 degrees left with speed 3". "Move camera with speed level 3 towards the position of 180 degrees" is an absolute command. The coupling of direction and speed level to one control command "Move left with speed 3" groups logically related issues together, and it alleviates the communication paths from the transmission of at least one additional message. Using such a command requires to give a command to stop the camera at the desired object.

In general the motor of the camera hardware possess not only the state "On" or "Off", it allows the division of a movement into various speed levels. Because of the inertia of the motors at the alteration of the speed level, we often experienced a certain problem: A sudden acceleration to the highest speed level in a very short period of time can cause a damage of the camera motors or demand at least a re-adjustment of the motors. This must be prevented. The introduction of ramps for acceleration and to brake down a camera movement solves this problem. This acceleration in the step by step mode avoids the damage of the motors.

Storage of Positions

The storage of positions is very specific to the used camera type. Some cameras allow to store positions by their special control hardware, others return values of their current camera positions and consequently facilitate a storage by software. Hence the handling of stored positions is also the duty of the camera driver.

The storage of a position means to save the values of the current camera position as well as values of the actual adjustment of the lens (zoom and focus). In order to address different already stored positions we introduced position identifiers, i.e names and numbers.

The management of these positions and the respective position names is performed by the camera driver at the server site. If a client wants to move to a desired position, then client transmits the accompanying control command and the position identifier via HCP to the server. The server tells the camera driver to perform the movement to the named position. Depending on the camera type and the used storage technique the camera driver determines the requires values for positioning and adjustment of the lens. These values and the identifiers of the positions are kept internally at the server in a position table and stored in file for availibility at any restart of the server.

The HeiCam Control Protocol (HCP)

The HeiCam Control Protocol is the protocol for the transmission of the control commands from the client to the server. Even though this protocol is not strictly time-bound the amount of transmitted data should be as little as possible. Therefore, all commands are executed at the server using ramps to accelerate and slow down.

HCP uses the Remote Procedure Call (RPC) which is suited for the development of client/server applications. The RPC is able to use UDP or TCP as lower layers of its own protocol. It is important to the HeiCam application that any command given by the user is executed exactly once. The reliability of TCP guarantees the fulfillment of this major requirement. The use of these standard protocols straight forward and simplified the development of the HCP. At the transmission of a command via HCP a data structure is sent which contains all informations needed for the execution of the desired control command.

The Audio/Video Subsystem

The delivery of the audio and video data from the camera server to the connected clients is the duty of a separate audio/video subsystem. This separation of client/server program and AV subsystem allows the reuse of already developed software modules and the independence of a specific audio and video format.

The BERKOM Multimedia Collaboration Service (MMC) supports joint working in a distributed environment. It allows users to share applications and to participate in audiovisual conferences from their workstation [1]. The Audio Visual Component (AVC) of the BERKOM project is interfaced by the HeiCam application. The AVC is based on HeiTS and establishes audio and video connections between servers and clients. There is one AVC on each server or client machine.

The Heidelberg Transport System (HeiTS) provides the ability to exchange streams of continuous-media data with quality of service (QoS) guarantees - where applications can specify the requirements they have for the transport service. To provide these QoS guarantees, the protocols of HeiTS are embedded into an environment which provides real-time techniques and resource management. HeiTS transfers continuous-media data streams from one origin to one or multiple targets via multicast. HeiTS nodes negotiate QoS values by exchanging flow specifications to determine the resources required - delay, jitter, throughput and reliability. Therefore, the cooperation of HeiTS and the AVC as a separate audio/video subsystem offers a powerful and already available facility for the transmission of multimedia data streams.

The AV subsystem has a clearly defined interface, the Source and Sink Control Protocol (SSCP). It can be used to establish transmissions of continuous-media data. This protocol controls how to open and to close connections by definition of endpoints.

At the HeiCam application the camera server works as central control for the distribution of the audio/video streams. That means that each client is able to request the transmission of the video images at the server indenpendent of the control of the camera.

Implementation of the User Interface

The graphic-oriented user interface of the HeiCam client is the control panel of the remote camera control. The integration of all required functions and a clear arrangement of the control elements ensures easy handling and fast interaction with the application.

Buttons for each direction of horizontal and vertical movement and for zooming and focussing the camera are integrated into the control panel. A sliderbar for fixing the desired speed level of the movements is integrated as well. A click with the mouse on a button starts the movement in concerning direction. A further click stops the started movement. The click on the stop button executes the halt of all camera movements. Using the sliderbar to change the speed level during a moving operation is possible as well. Stored positions can be added, renamed and deleted with buttons. A listbox gives a summary of all current stored positions. Using the mouse by clicking into the listbox executes the movement of the camera to the desired position.

Besides the traditional user interface with buttons for each direction of horizontal and vertical movement and for zooming and focussing, our novel approach [5] means to a remote camera by clicking in the window where the video of the camera is displayed. The direct interaction on the video window follows human intuition: We experienced this to be most effective and easy to use. This approach contains two variants for controlling the camera by clicking with the mouse:

- The user issues a double click at an object in the video window. Subsequently the camera is moved to a position, where this object results to be in the center of the video window.
- The user issues a single click at one side or at a corner of the video window. The nearer the click occurred at the border of the window, the higher is the speed level of the movement. I.e. a click in the upper right corner of the window causes the camera movements "left" and "up", a click nearby the center of the window causes a slowly move.

5 Experiences and Outlook

Since the start of the HeiCam project and its development in 1991 different HeiCam versions on two platforms were implemented. They were shown at many exhibitions around Europe including Cebit '92 at Hanover, Germany and the Security '93 at Utrecht, Netherlands. The field of the security and surveillance industry is changing more and more towards digital processing. Therefore, the HeiCam application finds more and more interest.

A **script language** allows to record and to playback sequences of previously issued control commands. The supervisor is able to concentrate on the monitoring only. In order to record a sequence the controller has to start the recording and to issue all commands as usual. All control commands and the time between two commands is kept in a editable file. A script command to restart the programmed macro file also provides the playback of an endless sequence.

The combination of a digital remote camera control with a **set of sensors**, such as photoelectric beams or motion detectors, extends the automation of industrial monitoring systems. If an event occurred and is caught by the sensors, the camera would move directly to the scene and the video image is automatically shown at the controller's monitor and recorded to a file.

The **detection of motion** can be done by analyses of a digital video. Video compression techniques using interframe coding allow the detection of motions by an analysis of the size of these delta frames. A sophisticate analysis of a live video may also be done to control a camera by tracing a desired moving object.

HeiCam comprises the remote camera control and interfaces the transmission of live video. It includes the possibility to store and retrieve the video images in a simple way. The development on different platforms, the simple integration into existing applications and the sophisticated user interface opens a wide market and the access to a large number of PS/2s and workstations.

References

[1] Altenhofen, Dittrich, Hammerschmidt, Käppner, Kruschel, Kückes, Steinig: *The BERKOM Multimedia Collaboration Service*; 1st ACM International Conference on Multimedia, Anaheim, Ca., June 1993.

[2] Hehmann, Herrtwich, Schulz, Schütt, Steinmetz: *Implementing HeiTS - Architecture and Implementation Strategy of the Heidelberg High-Speed Transport System*; in 'Lecture Notes in Computer Science 614', Springer-Verlag, Heidelberg, Germany 1992.

[3] Lier: *Design and Implementation of a Remote Camera Control for a Distributed Multimedia System*; diploma thesis at the 'Fachhochschule Wiesbaden', Germany, April 1993.

[4] Sander: *Design and Implementation of a Remote Camera Control based on a Multimedia Transport System*; diploma thesis at the University of Mannheim, Germany, June 1992.

[5] Sander, Ralf Steinmetz: *Method and Apparatus for Controlling a Camera*; patent, International Application, according to the Patent Cooperation Treaty PCT/EP 93/02647

[6] Steinmetz: *Multimedia-Technologie: Einführung und Grundlagen*, Springer-Verlag, Heidelberg, Germany, September 1993.

[7] Steinmetz: *Data compression in multimedia computing - principles and techniques*, Multimedia Systems, vol.1 nr.4: 166-172, Springer International, Heidelberg, Germany, February 1994

[8] Steinmetz: *Data compression in multimedia computing - standards and systems*, Multimedia Systems, vol.1 nr.5: 187-204, Springer International, Heidelberg, Germany, March 1994

[9] Wieland: *Design and Implementation of a Remote Camera Control based on a Multimedia Transport System*, diploma thesis at the 'Berufsakademie Stuttgart', Germany, August 1993

Bernd Wolfinger (Hrsg.)

Innovationen bei Rechen- und Kommunikationssystemen

Eine Herausforderung für die Informatik

24. GI-Jahrestagung
im Rahmen des 13th World Computer Congress
IFIP Congress '94
Hamburg, 28. August – 2. September 1994

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest