MultiMedia Digital Conferencing: A Web-enabled multimedia teleconferencing system

by C. Bisdikian
S. Brady
Y. N. Doganata
D. A. Foulger
F. Marconcini
M. Mourad
H. L. Operowsky
G. Pacifici
A. N. Tantawi

The convergence of the telecommunication and information technologies brings a real revolution to the communications world and hence in the way that people interact and communicate with each other. With the development of the World Wide Web technology for browsing and navigating through the Internet, complemented by the global telecommunication network, new, easy-to-use tools are being developed to simplify and enhance collaboration among people. In this paper, we describe the MultiMedia Digital Conferencing (MMDC) prototype system, which is a Web-enabled teleconferencing system. The MMDC server allows heterogeneous clients to use the Internet and the telecommunication network in order to establish and participate in data and/or audio conferences through the use of a variety of audio systems and computing platforms.

1. Introduction

As the Internet, the global system of networked computers, becomes more pervasive in our everyday activities, so does the need to exploit its global reach. The Internet was initially conceived as a network over which researchers would exchange electronic messages. Yet nowadays, short electronic notes constitute only one of a myriad of traffic types traveling the Internet. Multimedia traffic encompassing graphics, images, video, and audio consumes the bulk of the Internet networking resources.

To cope with the increased demand for Internet resources and services, its backbone is being upgraded from slow voice-grade lines to super-fast fiber optic links. New communication protocols are also being developed, allowing time-sensitive traffic to be delivered to its destination in a timely fashion. Thus, the Internet is evolving from a mere narrowband transport means for message exchange into a full-fledged broadband-services network. In addition to the development of the Internet transport mechanisms, the rapid development of the

0018-8646/98/$5.00 © 1998 IBM

World Wide Web (denoted WWW or just Web) browser technology, which provides an easy, point-and-click approach for navigating through the immense amount of multimedia information scattered among millions of Internet-connected computers around the globe, has brought the riches of the Internet to the desktops of millions of people, with no more than a modem-equipped personal computer [1].

A natural consequence of this easy access to a global information infrastructure is multiparty, multimedia collaboration and interaction, with Web browsers as interfaces. The idea of integrating multimedia and collaboration is not a new one. Indeed, Vannevar Bush's visionary description in 1946 of MEMEX [2], a networked, terminal-like device, combined multimedia and collaboration. In 1968, the idea of multimedia conferencing and screen sharing was demonstrated in [3]. Furthermore, there now exists a well-established field of research, computer-supported cooperative work (CSCW), focusing on human groupware and the use of computer technology to support collaborations [4–6]. A variety of groupware systems have been developed, ranging from text-based chat tools, electronic whiteboards, and video conferencing to document and application sharing and virtual classrooms and meetings.

Yet, it can be argued that the multimedia collaboration system in greatest use today is a combination of audio conferencing and fax, in which pages are faxed between meeting participants. However, advances in computing and networking technologies will result in multimedia collaboration systems with which people can share information and interact through audiovisual interfaces. Different types of multimedia collaboration systems have been developed [7], such as Touring Machine [8], DECAF [9], DistView [10], scientific collaboration [11], interactive distance learning [12], and groupware toolkits [13]. An architecture for a real-time multimedia collaboration system based on the T.120 standard [T.120] is presented in [14]. Commercial examples of multimedia collaboration tools are Microsoft's NetMeeting [Microsoft], Intel's Proshare [Intel], and DataBeam's Conference Server [DataBeam].

Multimedia collaboration systems may be broadly categorized into two architectures: centralized and distributed. In a centralized collaboration system, the control of group collaboration resides in a central-server complex. Such a server processes requests from group participants, manages the distribution of information to the group, and controls the flow of information within the group. The centralized architecture leads to a client/server arrangement, in which the client code runs on the workstation of participants and the server code runs on the centralized server. In a distributed collaboration system, the control is distributed among the computing equipment of the individual group's participants in a peer-to-peer arrangement, in which common code runs on the workstations of participants. The centralized approach benefits from the simplicity of implementation, operation, and management but is challenged by performance and reliability. The distributed approach saves the cost of a central server but is quite challenged by synchronization control and management.

The nature of integration of the multimedia collaboration system with the computer equipment of a participant has a direct impact on user acceptability, ease of use, and ultimately product deployment. Factors affecting such an integration are the need for user installation and management of software, user familiarity with the visual interface, and the type of user computer equipment. The multimedia collaboration systems mentioned previously require a software package to be installed and managed by the user. Furthermore, different collaboration systems provide different, unfamiliar user interfaces. An alternate approach is the use of Web browsers as user interfaces and the integration of the multimedia collaboration system with the Web browser installed on the user computer equipment. This approach is becoming the new trend in the development of multimedia collaboration systems [15–18]. Capitalizing on recent advances in Web browsing technology, the standardization of the Hypertext Markup Language [HTML] (HTML) for hyperdocument publication, and the introduction of the Java** language [Java], we developed the MultiMedia Digital Conferencing (MMDC) system, a Web-enabled prototype conferencing service that is centralized and fully integrated with the Web browser.

In a typical MMDC session, a participant uses his Web browser to visit an MMDC home page similar to the one illustrated in **Figure 1**. The operations available to the participant are divided into four groups: scheduling and reserving a conference, joining a conference, managing data content for a conference (in the form of "slide trays"), and obtaining customer support. After reserving a conference, the participant joins the conference using his Web browser. The telephone acts as the audio interface to an audio bridging system (combining several audio signals). The telephone number specified in the "join-request panel" will be dialed by the system. Upon answering, the participant joins the conference in both the audio and data domains.

Our objective is to marry the user-friendliness and pervasiveness of Web-based multimedia browser interfaces with on-line interaction and collaboration, using text, graphics, and voice communications. The MMDC prototype is a collaborative client/server system that has been guided from its conception to its realization as a working prototype by stringent requirements. Namely, the system must be Web-based (i.e., using Web standards for

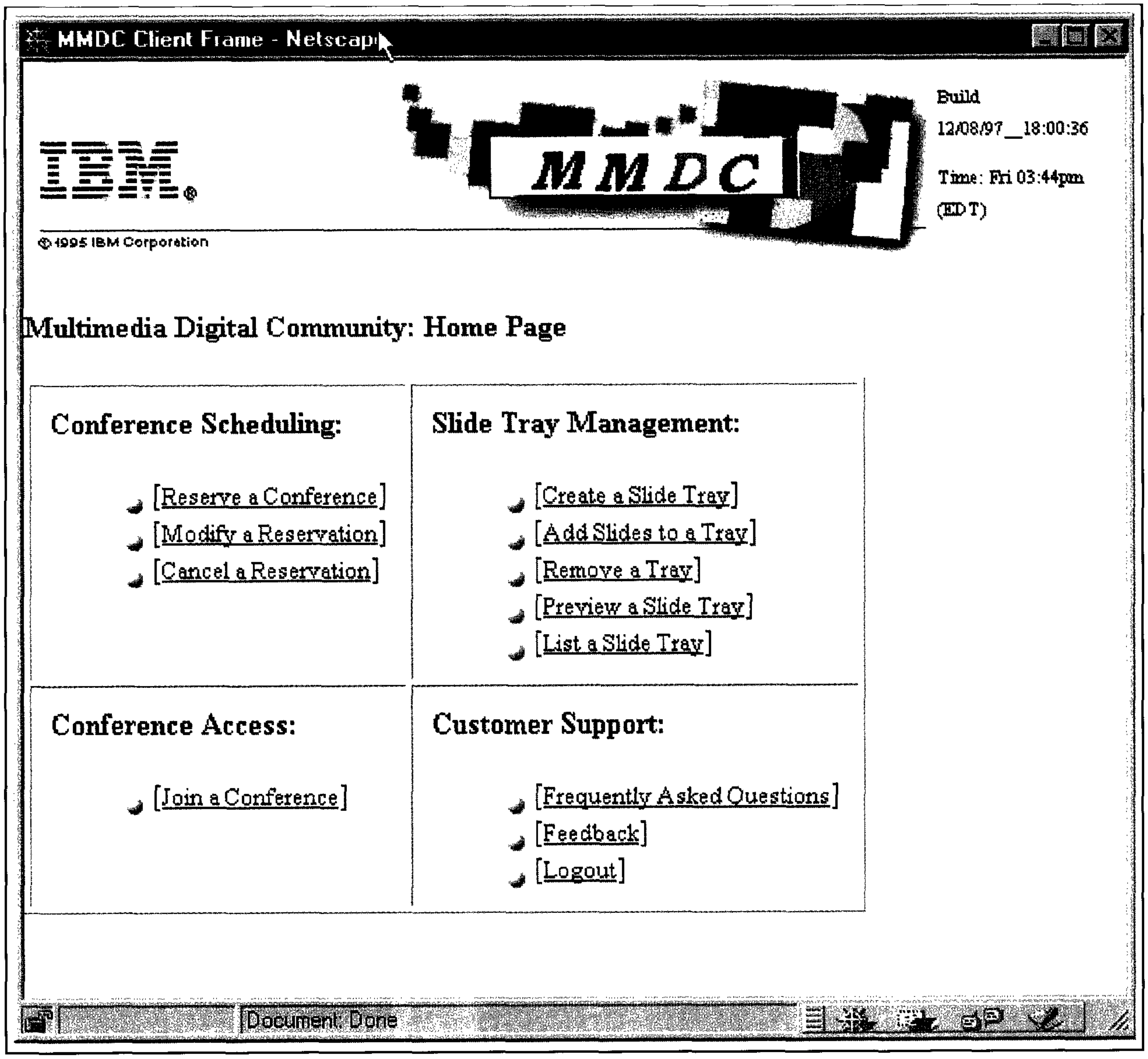

Figure 1

MMDC home page.

information transport and content representation), platform-independent (i.e., easily migratable to different user-client platforms), and scalable on the server side. Also, the system must have high-quality audio.

In this paper, we describe the MMDC system and its components. The paper is organized as follows. In Section 2, we describe the MMDC system and its objectives. In Section 3, we provide a description of the MMDC functions. In Section 4, we present the MMDC architecture and elaborate on the key software modules in our working prototype. In Section 5, we compare MMDC with other audio-/data-conferencing systems and expand on its application range. We conclude in Section 6 with a summary of our work and its present status.

2. The MMDC system

The MMDC project started out as an exploration of real-time, multimedia, collaborative applications that could be supported in a broadband environment. As indicated in the introduction, the phenomenal success of the Web gave us the confidence that standards for the format of multimedia content were being achieved. Moreover, we were interested in creating a solution that would be of interest to the customers of the IBM Telecommunications and Media Industry Solutions Unit.

Therefore, our goal is not merely to create an application in which individual Internet users can collaborate in peer-to-peer relationships, much like today's available products [19]. Rather, our goal is to provide an infrastructure by which existing businesses can legitimately exploit the growing power and popularity of the Internet and intranets[1] to enhance their current business products. A client/server architecture for the MMDC system serves this need much more effectively than a peer-to-peer architecture. For example, in our first application, the integration (and enhancement) of traditional audio-conferencing with document conferencing, our goal is not simply to create software for each of the conference participants but to design a complete system by which telephone companies (TelCos) and other conference-service providers owning an MMDC server can support and sell enhanced conferencing services to their customers, the conference participants.

The selection of our first application is highly motivated by the great market interest in collaborative teleconferencing solutions. This market area has been growing at a substantial rate over the last few years, and market analysis reports forecast continued growth, at an annual rate of about 33%, for at least the remainder of this century [20]. Additional application areas for MMDC include distributed multimedia classrooms, town-hall-type meetings, Internet-based group electronic commerce (e.g., shopping in groups), and audio on-line services.

Based on this perspective, the design points for MMDC are as follows:

- *Client heterogeneity* The system has to support several forms of heterogeneous environments. Participants will naturally expect that any service sponsored by the TelCo will be as universal as telephone service itself. Thus, we have to support a heterogeneous set of participants within a single conference. Participants should not worry about each other's Internet platforms before scheduling a collaborative teleconference any more than they worry about each other's phone equipment or long-distance carrier before scheduling a conference call. In addition, we do not want to burden a conference-service provider, for example a TelCo, with supporting various levels of platform-specific code. Thus, the decision was made that all code destined for execution on a participant's platform be written in the *Java* object-oriented language [Java]. Also, there are no predefined dependencies of MMDC on any particular server platform. Although the initial implementation of the MMDC server was developed on the IBM AIX* operating system, there are no inherent system dependencies.
- *Application flexibility* The MMDC system has to evolve along with the Internet. Thus, the underlying architecture has to allow easy extension to support new Internet technologies as they emerge. The system architecture has to permit applications to replace one another or interact with each other during the course of a conference, depending on the type of conference.
- *Server scalability* The architecture has to scale with the expected growth in Internet usage and function. We expected, of course, that we would easily obtain some scalability by off-loading as much function as possible onto the participants' computers. But we also know that our server must scale as well and grow into a heterogeneous network itself as the conditions of the individual MMDC customers, e.g., the TelCos, warrant.
- *Participant ease of use* Our ultimate goal for MMDC is to be as simple to use as the telephone. To this end, most interaction with the MMDC server is through the user's own Web browser. Ease of use also extends to installation and configuration of participant software. Our plan is that participants do not have to preload or install any MMDC client code prior to using it. The Java operating environment achieves this objective by dynamically downloading Java applets that can run locally within a Java-capable browser. Moreover, the client code does not require any configuration on the user's part.

Figure 2 shows the MMDC system configuration, where users at business or home locations connect to the MMDC server through a variety of means. There are two logical paths to the MMDC server: one is for voice communication, and the second is for data communication. Currently, the voice connection to MMDC is through the public-switched telephone network (PSTN), although other networks are under consideration as well. For data communication, the logical path is the Internet, to which a user may be connected directly via the office local area network (LAN) attachment, through a gateway (GW), or indirectly via a modem connection to an Internet service provider. The Internet makes available to its users the wealth of information stored on Web servers. An alternate data-communication path might be through a corporate intranet. The MMDC server provides all of the necessary facilities for audio and data conferencing. Note that voice-only participants may be allowed to access some of the MMDC functions; e.g., MMDC may call and admit a person who has only a telephone set into a conference.

Figure 2 depicts an MMDC with independent voice and data networks accessing the MMDC server. Although this is not a requirement of MMDC, which can operate over a

[1] An intranet is a private data network that uses the Internet technology for information exchange. In this paper, we do not subsequently distinguish between an intranet and the Internet.

single network, with both voice and data sharing either the PSTN or the Internet, the dual-network solution is the result of input from customers (mainly TelCos) and feedback from actual users of MMDC, who prefer the ability of PSTNs to deliver high-quality voice communication. The Internet has yet to support the quality of service needed for reliably delivering real-time voice traffic. Moreover, in an office environment, where MMDC would most likely be used first, the dual-network solution is a natural choice, as most office computers are connected to the Internet via high-speed LANs not associated with the private branch exchange (PBX) and PSTN connections available in the office. (The overlay of the telecom and Internet networks in the figure represents the use of the telecom net by the Internet.)

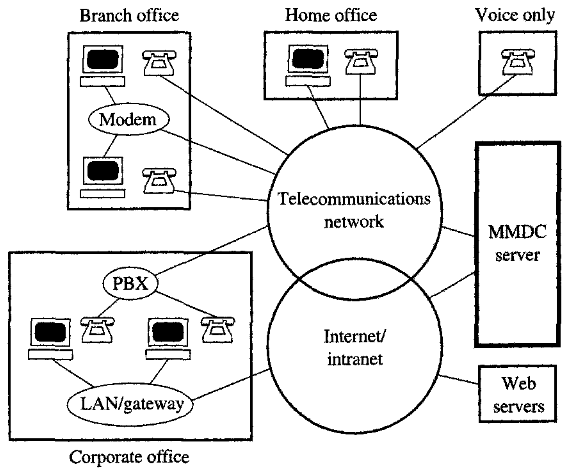

Figure 2

MultiMedia Digital Conferencing system configuration.

3. The MMDC functions

The integration of voice and data conferencing in MMDC provides an effective alternative to the type of teleconferences typically held today, for which conference materials are faxed to all conference participants prior to the conference. From the outset of the MMDC project, the intent has been to simplify and improve the effectiveness of conference calls by exploiting the multimedia capabilities of the Web. In particular, we eliminated the need for delivering any presentation materials to conference participants prior to the conference and simplified the process of distributing additional material during the conference, should the need arise. All interactions occur through both text and graphic data files over the Web in real time, but also through actual voice communication.

The MMDC user accesses, through a Web browser, all MMDC functions by first opening the Uniform Resource Locator (URL) for the MMDC home page (shown in Figure 1) and identifying himself. Then, the user is presented with a menu of all MMDC functions. Examples are reserving time slots for conferences, changing conference reservations, selecting and joining an ongoing conference, reviewing slides, and reviewing a conference transcript or previously shown slides. The MMDC server, at a conference participant's request, establishes voice conferencing with all or some of the members of a conference by calling the participants' phone numbers. This is in contrast to typical audio conferences, where the participants call the conference center and provide conference passwords. MMDC also allows voice-only conference participation through a call-in number. A participant may control the audio portion of the conference by clicking on audio control buttons on the conference Web page. (Functions such as disconnect and redial are already implemented; others such as volume control are planned for the future.)

For each conference, a special participant is identified as the conference chairperson, who moderates the conference. The chairperson or a presenter is able to preview and edit slide presentations prior to broadcasting them to the conference participants. The chair or a presenter is also able to send slides (preloaded data or just retrieved from somewhere on the Web) to all of the participants, thus ensuring that all conference participants have a common view and perception of the current state of the conference. The conference chairperson is able to dynamically disable or enable the addition of new conference participants, or dynamically disable or enable audio or data communication with conference participants. Furthermore, the conference chairperson controls who may take the "floor" (i.e., permits a participant to push slides to other participants and control other similar functions). In this fashion, conflicts due to competition among participants to simultaneously control conferencing functions do not arise.

Fundamentally, the key objectives of any conference are the creation and delivery of information to be shared among the conference participants. In MMDC, these objectives are accomplished in the following manner:

- *Creation of the presentation* A conference participant is able to load his or her presentation, through the use of the MMDC slide-tray management functions, by identifying the necessary files on his or her local disk. Currently, we require the presentation to be in HTML format. We expect to eventually support all major presentation formats, at least through the use of filters that convert files in other formats to HTML files. After the presentation slides are accepted and stored by the MMDC server, the participant is able to preview them and make any necessary changes. The participant may continue to review and revise a presentation, even during the conference. Further, the participant may decide to provide access to the presentation material to others for updating or reviewing.

- *Delivery of the presentation* Collections of presentation material, which we call *slide trays*, are created independently of any conference. The chairperson gains access to a slide tray by asking for it when he joins the conference. The chairperson is then able to view the list of slides by clicking on a button in his window and to cause files selected from the list to be broadcast to all participants in the conference.

In order to provide each presenter with the full capability of the Web, HTML is the presentation format of choice. If any presentation contains a URL reference, the presenter, by simply clicking on the reference, can cause everyone to visit that site—an activity we call *group surfing*. We also allow the presenter to start group surfing by specifying any URL during a conference. Of course, the presenter may choose to display a page from the slide tray at any time.

4. MMDC system description

MMDC is a client/server application, with code executing on each participant's workstation as well as within the MMDC server. One of the key architectural ideas upon which MMDC is built is the universal availability of its functions. The selection of HTML for content preparation and display has its advantages as well as limitations. One concern is the stateless nature of the hypertext transfer protocol [HTTP] (HTTP) [21] used for the transport of HTML-coded content over the Web. In other words, every time a page is requested from an HTTP server, the requester must create a new link with the server, regardless of the number of times that the requester may need to access data on the server. For a collaborative system in which requests will be sent repeatedly to the same server, this continual reestablishment of connections represents a significant amount of overhead. One of the proposals for extending the HTTP protocols to allow synchronous collaboration would permit connections to be maintained for the duration of a client's interaction with a given server. A second concern is that for a conference to proceed smoothly, data must be delivered to the clients simultaneously. If the clients are required to use the standard HTTP protocol to open a connection with the data source for each new page of content, requests from the conference participants will arrive in bursts, with the resultant possibility that some client requests may be refused, causing a delay in that client receiving the next page and ultimately delaying the entire conference. Clearly, the larger the conference, the greater the likelihood of this occurrence.

In the rest of this section, we present the key software elements of the MMDC server and client, and we demonstrate how the MMDC architecture overcomes these two problems.

- *The MMDC server*

The MMDC server, as illustrated in **Figure 3**, is divided into a number of components:

1. A concierge, which takes on many of the functions associated with a human operator in contemporary audio teleconferencing systems—for example, client authentication and conference scheduling. This registration-desk service is the first point of contact of a client with MMDC and is accessed via a Web browser.
2. A conference-control capability for managing membership in the conference and broadcasting conference materials to participants. Conference control allows a chairperson to control who can present information and select content for display. The multiple conference-control blocks in Figure 3 represent multiple simultaneous conferences, as described below.
3. A Web processor, which is responsible for all slides and collaborative content presented during a conference. It provides such functions as creating slide trays from user input, obtaining multimedia content from remote Web servers, and preparing content for broadcast to the conference participants upon demand. These capabilities eliminate the need for faxing conference presentation material among the conference participants. This is an example of what we refer to as a *collaboration processor*.
4. An additional set of collaboration processors for allowing collaborative services, such as whiteboards, to be attached to the MMDC server as needed. Other services, such as shared file cabinets, bookmarks, and user profiles, may be implemented in the future.
5. An audio conference center for the implementation and control of the audio portion of the conference.
6. A server-management component for a variety of functions, including resource management, data management, transcription management, and operations and service management.

Other modules may exist that can be used off-the-shelf and plugged into this architecture to deliver additional services, for example, a transcription manager or a billing subsystem. The core of the MMDC server development, though, has been focused on the following four modules: concierge, conference control, Web processor, and audio

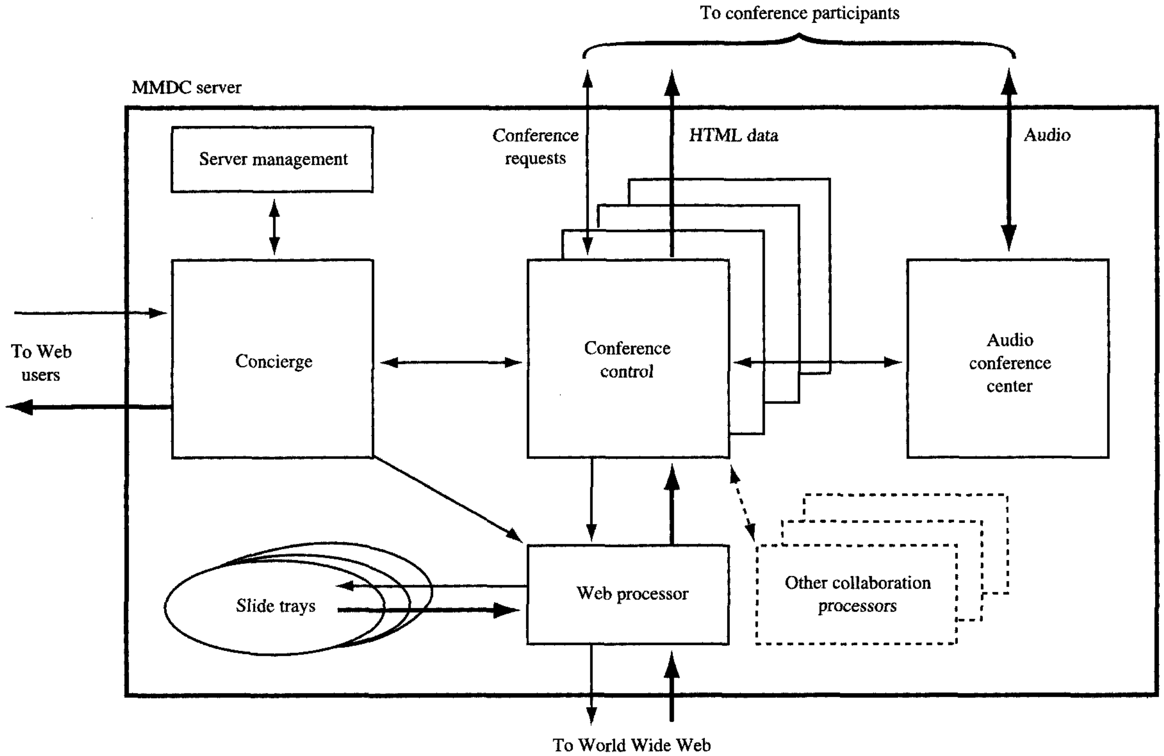

Figure 3

MMDC server components.

conference center. Each individual conference will draw on the services of all four components (but involve the concierge only initially, as described below). Finally, recall that the collaboration processors, which can also be considered as part of a given conference, are installed in an MMDC server, as needed, to add enhanced collaborative applications.

The concierge
As the name suggests, the concierge is responsible for managing access to the MMDC server. All functions required by a user prior to actually joining the conference are handled by the concierge. The concierge is also the focal point for all general services required by all conferences, such as reservation and resource management, maintenance of user data, initializing client workstations by downloading Java applets that provide MMDC client functionality, interaction with systems operation personnel, maintaining system logs, providing system recovery, and establishing connections to outside data sources such as enterprise billing records.

The concierge interacts with clients via a standard Web browser and Web server, using standard Common Gateway Interface (CGI) scripts [22]. When a conference is scheduled to begin, the concierge creates that conference by starting an instance of conference control to manage the conference. When a client joins a particular conference, the first contact is with the concierge, which performs an initial security check, and then returns the necessary information to the client to allow it to cancel the client–concierge connection and create a persistent connection between the client and the conference control instance managing the requested conference. All further client interactions with the server are handled by the conference control.

The conference control
Each conference session takes place and is managed under the auspices of conference control [23]. The conference control module plays two key roles in delivering the type of collaborative applications envisioned for MMDC. First, it is this component that joins participants into

collaborative groups and maintains the status of communication channels needed to attach those participants to the services of the collaboration or conference. As part of this function, conference control ensures that information about the participants in the group is forwarded to each of the participants, giving them explicit listings of group membership. Second, it is the central point of control for access to the application services that make up the collaborative environment. In business audio conferencing with Web document sharing, these services are the assembling of HTML content to deliver to conference participants and various sets of controls over the audio environment. Conference control can enforce policies governing which participants can access these services and when.

There are several reasons why we chose the approach of creating a separate instance of conference control for each conference:

- *Scalability* By managing conference sessions separately and eliminating platform dependencies within conference control, the MMDC server can scale with demand simply by adding additional processors. Furthermore, if telephone-switching hardware is distributed, conference control can be placed on the processor that can communicate with the telephone hardware most efficiently.
- *Security* By having each conference session managed by a separate entity, we can more easily guarantee the security of data within each session. Participant clients deal with only a single conference control associated with their session, and instances of conference control never communicate with each other.
- *Implementation* It is easier to implement a conference control function that focuses on managing only one conference.
- *Stability* Problems that arise in one conference are less likely to affect other conferences.

We envision conference control taking different forms as different collaborative applications are created; however, on the basis of our experiences in creating the initial conferencing application, we believe that all forms will maintain a similar general structure. A generic conference control component comprises

- A *communication* component: All accesses to data outside the MMDC environment are handled by a single communication component, which provides a general abstract interface to the rest of MMDC. The primary reason for this approach is to eliminate platform dependencies. For example, conference control currently executes only on AIX platforms, where it uses the AIX version of the UNIX** socket library to access the Internet suite of communication protocols, but it could be ported to a Windows NT** environment by changing the communication component to use the Winsock interface instead. None of the internal conference-control function would have to be changed. Similarly, the communication component hides the precise nature of our telephony hardware, so changes needed to allow MMDC to support new hardware can be isolated to the audio layer within the communication component. I/O requests handled by the communication component include audio-port assignments to newly joining clients, message broadcast to all members of the conference, message transmission to a selected group of participants, maintenance of the incoming-request queue, and monitoring of the audio and data channels to handle hardware failures.
- A *membership* component: The membership component handles those issues related to attaching a new client to the conference. In our case, the primary issues are ensuring that the persistent data and audio connections are made and remain intact throughout the conference. Similarly, at the end of the conference, or when a participant decides to leave, the membership component is responsible for recovering resources.
- Several *collaboration-processor* components: Our intent is to develop a conference manager that will be easily configurable to meet the different requirements of various collaborative activities, services, and resources that users wish to include in their on-line collaborative environment, e.g., a shared whiteboard and access to directory services. We achieve this requirement with the addition of collaboration processors. As mentioned above, the Web processor is an example of this type of component.

Conference control defines an object class for collaboration processors to allow us to dynamically install collaboration processors during a conference as well as to access collaboration processors running on different platforms, further supporting platform independence and scalability.

The Web processor

The primary feature of an MMDC data conference is MMDC's ability to immediately deliver content to all participants simultaneously, upon demand by the conference chairperson. This feature is implemented within the Web processor.

The name *Web processor* stems from the fact that all content presented in a data conference is coded in HTML and is transmitted to and from the conference participants by means of HTTP. These are the primary means by which all data is transmitted to an individual's browser attached to the World Wide Web. Each person who participates in an MMDC data conference uses his or her own browser, without needing to change any of the browser's options or configuration. MMDC thereby allows users to exploit the full power of the Web during the conference. For example, we allow group Web surfing, and we allow individual participants to branch off and take separate trips through the Web. Most content presented, however, will come from collections of HTML files preloaded onto the MMDC server platform by the individual participants.

In order to address the HTTP performance concerns previously mentioned and enhance MMDC's scalability, we provide a server "push" model rather than the standard client "pull" model employed on the Web. Whenever a conference chairperson makes a request for content, whether it is an explicit request for a slide in a slide tray or a URL on the Web, or an implicit request such as clicking on a link in the current page displayed, that request is transmitted to the Web processor. The Web processor parses the request, determines all files necessary to satisfy the request, obtains those files from either a slide tray or a remote Web server, and distributes them to the conference participants. Multiple files are generally required to satisfy a request because of the syntax of HTML itself. For example, graphic images displayed in a Web page are not embedded within the HTML file for the page. Rather, the HTML file contains a reference to a separate file that contains the image. Then, as a typical Web browser processes the HTML page, it makes a separate request for each image file referenced. Java applets are obtained in the same way. The approach used by the typical browser is an example of the client pull model. Had we decided to use the client pull model for MMDC, each conferee's browser would make a separate request for each file. If a file were located on a remote Web server, the number of such requests from a large conference could overload the Web server, resulting in a number of requests being rejected, delays in obtaining the content, and disruption of the conference.

To accomplish its mission, the Web processor performs the following functions:

1. *Obtain the requested slide or URL.*
2. *Parse the requested file:* In the vast majority of cases, the requested file will be coded in HTML. In some cases, especially when the HTML "frame" feature is used, some of the referenced files will themselves contain HTML. In those cases, the Web Processor must parse the newly obtained HTML files and obtain any additional referenced files.
3. *Broadcast all required files to conference participants:* As indicated above, the key to meeting our performance and scalability objectives is providing a server push model. In the standard client pull model, each participant's browser would request each referenced file from a remote file server (the MMDC server for slides, or an arbitrary Web server for other Web content). In our server push model, the Web processor makes a single request for each file, on behalf of the entire conference, and then broadcasts it, together with control information used by the MMDC client software, to the conferees, using HTTP just as a typical Web server would. The client-control software on each participant's platform receives the files and makes them available to the browser.

However, in order for the user's browser to request the files from the client-control software as well as support other MMDC functions, changes must be made to each HTML file before it is delivered. The following items describe those changes.

4. *Modify all references to external files:* Part of the external file reference within an HTML file is the name of the Web server that contains the file. In our server push model, the browser obtains all necessary files from the MMDC client-control code running locally on the user's workstation. To accomplish this, without changing the user's browser, the Web processor changes all file references in all HTML files to refer to the local MMDC client control instead of the external server that really maintains the referenced file. File names are also changed to handle the problem of two files on separate servers having the same names. If the file names were not changed, the two files would be indistinguishable once we replaced the real Web server name with the client control's name, and the browser would receive the wrong file.
5. *Modify all HTML anchors:* A major contributor to the popularity of the Web is the simplicity of accessing a Web site simply by clicking on a link in a Web page. We earlier referred to this as implicitly requesting a page. Such links are represented by HTML constructs called *anchors*, each anchor containing the URL for the given site. In order to support group Web surfing, conference control must intercept each implicit request and forward it to the Web processor. To accomplish this, the Web processor initially converts the URL for the site in each anchor into a Web processor request for the URL. When a conferee clicks on a link during the conference, the modified anchor causes the client control to intercept the request and transmit it to conference control, which then delivers it to the Web processor.
6. *Process user-generated slides:* Since the primary function of a conferencing system is allowing the participants to share their ideas, we expect that the preponderance of data requested will be HTML files created from users' presentations. Desktop presentation

packages such as Microsoft PowerPoint and Lotus Freelance Graphics are already capable of producing HTML. Conference slides have all the hyperlink functionality possible in Web content. In particular, slides can have links referencing remote Web sites as well as other slides in the slide tray, and the chairperson can display slides and Web pages in any order during the conference. Prior to an MMDC conference, users preload their presentation files into the appropriate slide tray on the MMDC server through a function provided by the concierge. Immediately upon receipt of an HTML file, the Web processor is called to process it in much the same manner as described above. This is done, of course, to minimize response time during the conference. One difference is that external files referenced in the HTML are not retrieved until the slide is requested during the conference. However, the actual content delivered to the client in response to a slide request is structurally identical to that delivered in response to a request for a standard URL.

The audio conference center

In order to create and manipulate the audio domain, conference control invokes the services of an audio conference center. This center manages the audio conferencing component of a multimedia conference and employs an audio-conferencing hardware system to perform the *audio-bridging* function (combining multiple audio signals to produce their effective sum). The audio conference center is designed to support a variety of audio-conferencing hardware systems and is flexible enough to support new hardware systems without altering the rest of the MMDC server. Consequently, conference control is shielded from the details of the audio-conferencing implementation, helping to realize some of the network-technology independence that is a requirement placed on our overall system design.

Various audio-conferencing systems are being considered for MMDC: IP (Internet protocol) conferencing, PSTN (public-switched telephone network) conferencing, open-switch conferencing[2], PBX (private branch exchange) conferencing, and CTI (computer telephony integration) PC-based conferencing. Both IP conferencing and PSTN conferencing have been considered and rejected for the reasons described below. Currently, the audio conference center supports open-switch conferencing, PBX conferencing, and CTI PC-based conferencing.

[2] Unlike traditional closed-architecture switches, for which the programming interface to the hardware is proprietary, open programmable switches provide open programming interfaces, thus allowing application development by independent software developers.

- *IP conferencing* Internet-based audio-conferencing tools allow users to send over the Internet packetized audio messages from their multimedia desktop PCs to an audio server. The audio server is designed to manage the audio conference by combining the incoming audio packets from the individual participants and broadcasting the resulting mixed audio packets back to the participants. The transport of the audio packets is performed by utilizing standard Internet protocols. The voice quality depends on many factors, such as the audio cards used by the participants, the audio encoding scheme, the real-time audio transport control mechanism, the number of network nodes between the audio server and the participants, and network congestion. For real-time voice communication, the IP conferencing approach lacks good quality, as confirmed from our tests, and does not scale gracefully.
- *PSTN conferencing* Audio conferencing may be realized in the PSTN by means of the regular audio-conferencing services provided by TelCos. The interface to such conferencing services may be achieved through network signaling such as ISDN, which supports conferencing operations. This solution is good for TelCos that provide conferencing services through their own PSTN switches. In this case, the MMDC audio conference center could be tailored to use the conferencing facility that the telephone company owns. Since this approach depends on the existence of a Telco audio-conferencing service, the MMDC prototype did not implement the PSTN conferencing solution.
- *Open-switch conferencing* Using an open-switch architecture with appropriate digital conferencing hardware provides a lot of flexibility in terms of switching control and management. Such a switch is typically controlled by a host, which manages the switch and may have an interface to the switch in the PSTN for out-of-band signaling (use of a separate telephone network for call setup and signaling). The MMDC prototype uses the IBM Multiple Service Platform (MSP) software [24] to control an open programmable switch.
- *PBX conferencing* In an enterprise environment, the PBX is a natural place for audio conferencing. The PBX technology has advanced from analog to digital audio mixing, rather than audio bridging. (Mixing requires digital signal processing hardware to process the audio streams, whereas bridging simply combines the streams without processing.) The MMDC prototype uses the IBM CallPath* server software [CP] to control a PBX.
- *CTI PC-based conferencing* A small-scale audio-conferencing system may be built using a PC with hardware cards interfacing to the PSTN and with conferencing hardware cards having digital signal processors for voice processing. Such an approach

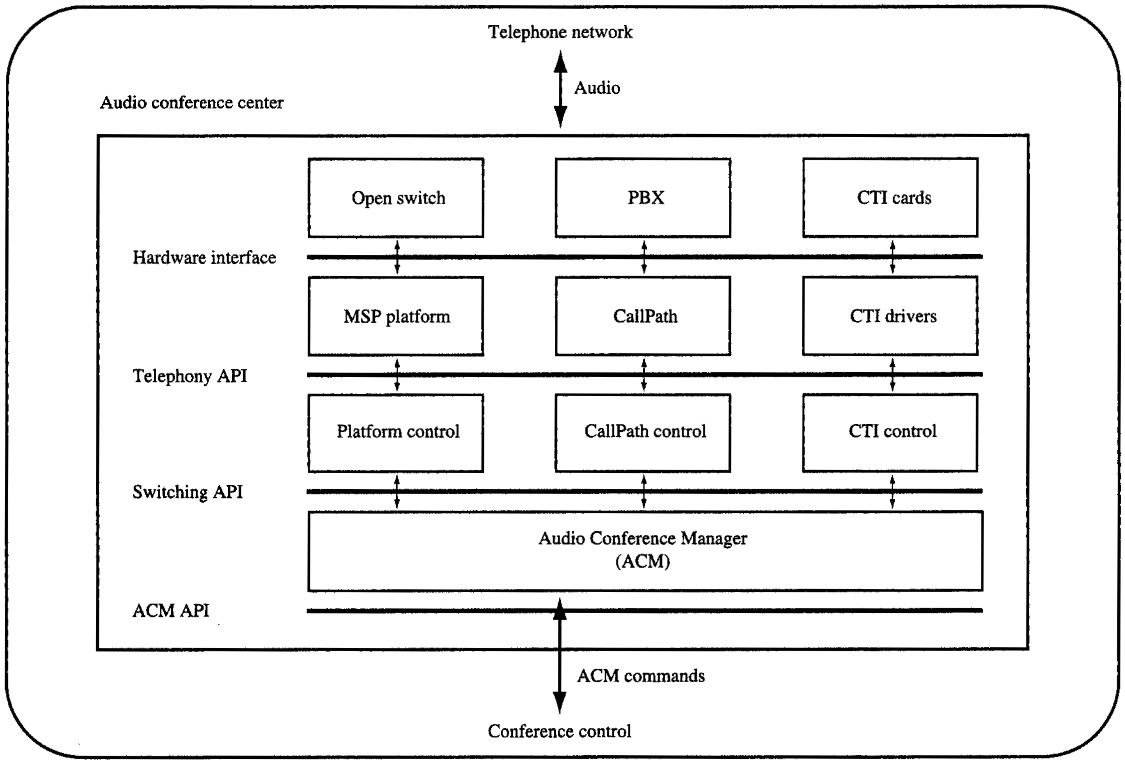

Figure 4

MMDC audio conference center.

requires the development of a software layer to control the CTI cards, through the use of drivers provided by the hardware manufacturer.

In order to support multiple audio-conferencing systems, yet preserve a common, generic audio-control interface to the conference control, we adopted the modular layered architecture illustrated in **Figure 4**. Depending on the audio-conferencing system, an appropriate manager (such as MSP, CallPath server, or CTI management layer) is selected. Associated with each such manager, we developed a controller, which abstracts different audio-conferencing systems to a common switching application programming interface (API). Then, the audio conference manager (ACM) acts as the layer between the conference control and the audio-conferencing system.

We chose the TCP/IP standard for communicating commands between conference control and the ACM, as well as between the ACM and the audio controllers. Thus, if desired, we could place the ACM on a machine that is separate from the MMDC server complex, the audio-conferencing hardware system, or both. This decision simplifies the development and implementation of the ACM. Thus, the ACM can be built independently of the core MMDC platform, and may be used for other applications that require audio-conferencing control capabilities.

In order to improve system performance, each MMDC user request to start a conference initiates an ACM server process, each such process handling one conference and managing the following important tasks: add or drop participants from the audio conference; respond to MMDC queries, such as the status of telephone connection and the status of the switching application; send queries to and receive responses from the switching hardware; poll the switching hardware for conference status; and send conference status messages to conference control.

- *The MMDC client*

MMDC allows participants to create and join conferences by accessing a given URL using a Web browser. A set of

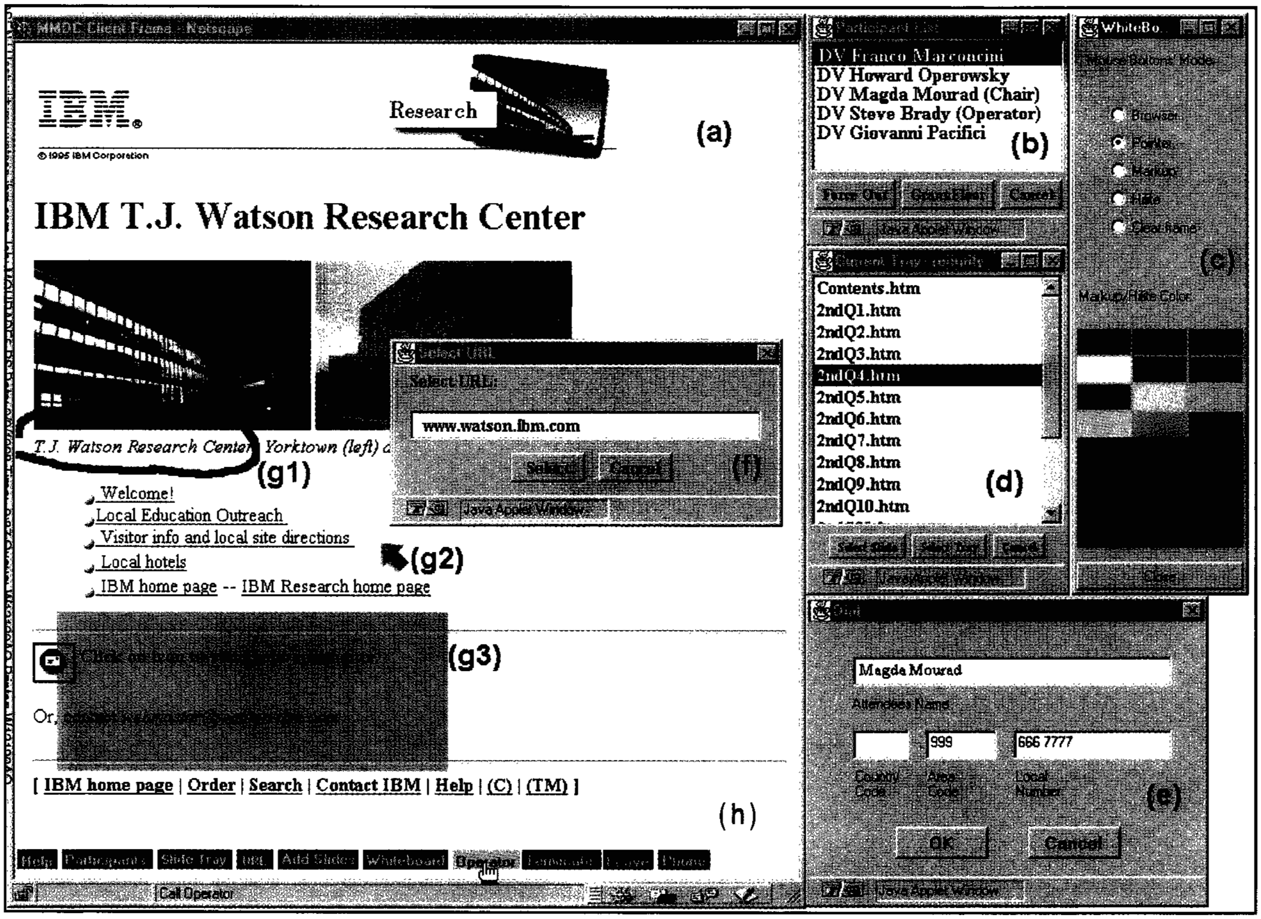

Figure 5

MMDC chairperson's user interface during a conference session: (a) Main browser window; (b) participant-list panel; (c) whiteboard; (d) slide-tray panel; (e) audio panel; (f) URL request panel; (g1) whiteboard markup; (g2) whiteboard pointer; (g3) whiteboard highlight; (h) control frame of browser.

menu-selection panels guides users through the reservation and joining processes. When a user joins a conference, the MMDC client (a Java applet) is dynamically downloaded. This applet establishes persistent communication with the MMDC server to allow participants of a conference to synchronously share HTML documents.

The MMDC client runs in three different modes: *chairperson*, *participant*, and *preview*. The mode is controlled by the MMDC concierge. If the user that joins the conference is the chairperson (the chairperson is designated at the time of conference reservation), the client is set to run in chairperson mode; otherwise the client runs in participant mode. If the conference is a preview session (i.e., a data-only session with only one participant), the client mode is set to preview. When in preview mode, the user can access only the subset of the client functions that concern the display of slides, adding slides to a tray, accessing and displaying URLs, using the whiteboard, and terminating the conference. When the MMDC client runs in participant mode, the user can access all of the preview-mode functions as well as control the audio and view the participant list. When in chairperson mode, the MMDC client can be used to control who holds the floor in the conference, add new audio participants, and disconnect active participants. **Figure 5** shows the screen display of the MMDC client interface during a conference session. The main window of the browser shows two horizontal frames: the *display frame* and the *control frame*. The display frame is dedicated to the display of the HTML documents shared by the conference participants. The control frame implements a set of buttons that allow users to access the panels shown, as well as a *help panel* and an *add-slides panel*. The control frame also has three additional buttons that allow users to invoke the help of a conference operator (*operator button*), leave the conference (*leave button*), and disconnect the entire conference (*terminate button*). The terminate button is visible only when the MMDC client runs in chairperson mode, while the participant and audio panels are not visible when the MMDC client is in preview mode. The help panel is used to display a context-dependent help message. The *participant panel* displays a list of conference participants that is dynamically updated as people leave and join the conference. The participant panel also displays information concerning each participant: the type of participant connection (Data and/or Voice), the participant mode (chairperson or participant), and which participant currently holds the floor. When the client runs in chairperson mode, the participant panel can be used to *grant* the conference floor and *disconnect* conference participants. The *display slides panel* (invoked by the "slide tray" button) allows participants to select slides for display as well as switch among slide trays. URLs can be accessed and displayed to conference participants with the *URL panel*. The *add slides panel* allows participants to edit the content of a slide tray during a conference. The *whiteboard panel* provides access to a markup application, which allows users to overlay a pointer, drawings, and text over the displayed document. The markup application can be used also to highlight portions of the document. The *audio panel* is used to control the audio session.

As shown in **Figure 6**, the MMDC client is composed of four main subsystems: *display engine*, *client control*, *conference cache*, and *audio*. The client control is responsible for sending commands to the MMDC server as well as receiving the HTML data for display. All of the HTML objects received by the client control are stored in the conference cache. In order that a single slide (or URL) be fully displayed, the HTML document, together with all of its components (e.g., GIFs), must first be stored inside the conference cache (according to the push model described in the subsection on the Web processor). When the HTML and its components are stored in the conference cache, the client control receives from the MMDC server a *display* command. Upon receiving a display command, the client control instructs the display engine to load the document stored in the conference cache. The display engine is the display component of the Web browser where the MMDC client runs. The client control is also responsible for receiving commands issued by the user. Depending on the specific action requested by the user, the client control may propagate a command to the MMDC server. Finally, the audio subsystem is the part of the client responsible for exchanging the audio data with the MMDC server. The audio subsystem is a logically distinct application that connects directly with the MMDC server and is controlled by the user and by the MMDC server. The audio subsystem can be implemented through

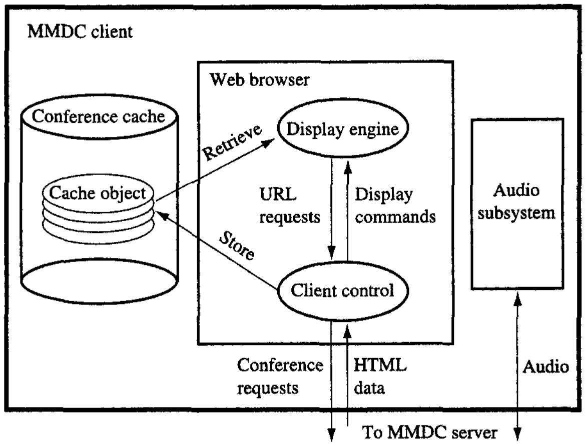

Figure 6

MMDC client architecture.

either an IP voice application such as *vat* [25] and *nevot* [26] or a telephone.

An important practical issue for the client is its ability to function properly in the presence of firewalls between the client and the server. For example, a server accessible through the Internet and a client Web browser behind a corporate firewall constitute a very common configuration. Typically, in such a case, the client Web browser is configured to use a proxy (or a similar provided functionality) to access servers on the Internet. One of the key features of the MMDC client is that it uses the Web browser to communicate with the MMDC server by means of the HTTP protocol, as opposed to managing its own communication channels to the MMDC server. Consequently, the MMDC client communicates with the MMDC server through firewalls in a seamless fashion, in the same way a Web browser communicates with Web servers.

5. Discussion

- *Competing solutions*

The principles and design objectives of MMDC, namely a simple-to-use, integrated data- and audio-conferencing application that is client-platform independent and scalable, sets MMDC apart from many other solutions in the collaborative-applications area, which require either homogeneous communication or homogeneous operating system environments. In this section, we present some of the other solutions offered. For an extensive collection of Internet-based collaborative applications, see [19].

Recently, there have been a number of trials conducted and offerings announced that overlap in various ways with

MMDC. Many such trials have been discussed on the Web sites of the parties involved. As expected, TelCos have been prominent in this area. Both AT&T [ATT] and Sprint Internet Conference Center (SM) [Sprint] have created Web interfaces to their teleconferencing services—interfaces that can be used to reserve or even control an audio teleconference. However, this control function is separate from the parallel data-conferencing solutions each provides. In the case of AT&T, the data and video conferencing are based on ITU standards (T.120 and H.320, respectively). In the case of Sprint, the data conferencing relies on platform-specific client software. In either case, it is assumed that in order to participate, clients should have downloaded and installed the prerequisite client code. While platform interoperability is ensured if the client software is available for multiple platforms, interoperability among different conferencing service providers or even among different standards-compliant (e.g., T.120) clients is not possible.

MMDC covers a richer set of connectivity options, including a two-network solution, which guarantees regular telephone quality for voice conferencing in addition to a rich set of tools for data sharing, Internet browsing in a collaborative manner, and whiteboard sharing. A simple voice-only conferee can also participate in MMDC sessions.

A single-phone-line solution is also possible (although not fully implemented yet) if voice is digitized and then packetized and transmitted as (Internet) data. A single telephone line, carrying both data and voice, can then be used for Internet access. Such users can participate in the same teleconference as two-network users. Although the quality of voice in a single-line solution is not as good as the two-line solution, because of bandwidth limitations, participants using two networks would still get the voice quality of a regular telephone line.

There are also a variety of Internet-collaboration tools becoming available that feature voice transport over the Internet [19]. Netscape's CoolTalk** [CT] and Microsoft's NetMeeting** [Microsoft] are among the more visible examples. However, since the Internet as it currently exists cannot offer the quality-of-service guarantees needed for high-quality voice, the MMDC use of the PSTN for voice transmission is an advantage.

Other solutions being offered by a variety of software companies are mostly built on products using T.120 and H.320 toolkits (see MCI/DataBeam [DataBeam], AT&T WorldWorx** 800 Personal Conference Service [WW]). Such solutions are based on the ITU standards to allow cross-platform interoperability and support, e.g., Intel ProShare** [Intel], Apple QuickTime** Conferencing [Apple], and DataBeam** [DataBeam]. They allow real-time interactive data, voice, and video conferencing. Further, they allow locations with dissimilar line speeds and video equipment to participate in a collaborative session. They can run over one (multiplexed) or more (separate) toll-free-telephone-number links with speeds ranging from 300 bits/s to 1.5 megabits/s. The TelCos offering these solutions provide these links and support services (e.g., quality guarantees, user assistance, and international access), but special hardware at the user's end and special networking links have to be leased by conference participants. In contrast, MMDC uses the existing PSTN and Internet connectivity of participants to deliver its service. These other solutions are not Internet-based, and reservations are made through an operator. The advantage of these solutions is that they offer the possibility of higher communication bandwidth, enabling users to have a high-quality video conference in addition to their data conference. The disadvantages, however, are the higher price as well as the special hardware and client software (platform-dependent) needed to allow multiple users to collaborate.

One of the objectives of developing collaboration systems compliant with the T.120 standard is to ensure interoperability between terminals, with no prior knowledge of the client systems. However, specific T.120 software must be installed for specific operating systems running on the client systems. Hence, participants exchange data, share whiteboard images, and transfer files from their geographically dispersed and possibly heterogeneous terminals. The same objective, namely interoperability between different types of client systems, is also achieved through Web browsers. In this case, however, common Java software runs on heterogeneous client systems. Nowadays, most computers connected to the Internet, regardless of their type and operating system, have Web browsers to exchange multimedia content by using standard protocols. Although the popularity of Web browsers has increased beyond estimations, no audio/data-conferencing service is available for Internet users that provides "audiographics" conferencing applications, with business-quality audio. In response to this need, MMDC was developed.

One of the most important distinctions between MMDC and T.120-compliant applications is the way audio is integrated into the data conference. MMDC interacts with the call managers of existing switching systems to set up audio conferences among the participants. In T.120-compliant solutions, however, the audio channels are handled directly by the Multipoint Control Unit, as defined in the T.120 standards. As a result, many implementors use a special T.120-compliant hardware component for bridging the audio channels, hence increasing the total system cost.

An alternative infrastructure for the implementation of interactive multimedia applications is the MBone (Multicast Backbone) Architecture [27]. This architecture uses a networking protocol feature, called IP multicasting, which allows the distribution of packets from a single source to multiple destinations, rather than the regular IP delivery to a single destination. IP multicasting constitutes the underlying technology for the MBone, thus making the realization of collaboration simpler. An example of a multimedia collaboration system using the MBone and the WWW is presented in [28]. The commercialization of the MBone is still under way.

- *Other MMDC applications*

While we are focusing, for our initial MMDC application, on what might be called Web-enhanced teleconferencing, the potential for the MMDC platform is considerably broader. A range of multimedia on-line services that integrate voice and data services can be readily envisioned and accomplished with existing MMDC components.

Among the most obvious additional applications of MMDC are those in distributed education. Using the existing MMDC system, an instructor can already convene a class, display lecture material, including Web pages, and interact with the class on the audio channel concerning that material. Students can use the existing MMDC user interface to indicate that they have questions, and the instructor can give the floor to individual students who have questions, are asked questions, or who are recognized in the course of discussion. Access to the vast information resources of the Web should be a major advantage in this setting. The combination of the Web and telephone should make it easier to bring guests into the classroom for special events, thus enriching the educational environment.

Another application of MMDC involves meeting formats that are currently done almost entirely face to face. Town meetings, school board meetings, judicial proceedings, committee meetings, and many others need not be restricted to those persons with the mobility to travel to those meetings. The combination of audio-conference access to meetings with the ability to view meeting presentation materials should enable remote participants to be brought into existing meetings. It is a small step, moreover, to conduct these meetings on an entirely distributed basis, where different participants may select one of several views (e.g., different slides) of the same meeting.

Yet another application, multimedia group shopping, seems to be an entirely new business opportunity. In multimedia group shopping, groups of individuals are able to shop together on the Web while interacting over an audio channel. Interesting variants of multimedia group shopping include scheduled shopping trips, perhaps involving friends, relatives, or others; public shop-and-chat sessions, in which an initiator can declare a shopping focus and anyone can join in; celebrity shop-and-chat sessions, in which a celebrity takes people on a shopping tour; and electronic auctions, in which the main browser window displays the item at auction, an MMDC-server auction application accepts electronic bids across the network, and the interaction is dominated by the auctioneer's patter, discussion of new items, etc. This is an important aspect of the shopping experience that has so far attracted relatively little attention among those who are developing electronic commerce for on-line services and the Internet.

Even more applications can be envisioned with additions such as the inclusion of the intelligent agent technology, voice recognition (even on a limited basis), and automated transcription. The components of MMDC can also be recombined in interesting ways to enable additional applications. The audio conference center, audio services, and the concierge can be recombined, for instance, to allow on-line conference-call reservations and audio conference control in audio teleconferencing environments, potentially reducing the cost of operators. Audio services and conference control, moreover, can be combined with Web access to enable multimedia mailboxes that provide combined access to voice mail, e-mail, fax, and multimedia messages.

6. Summary and status

We have presented the MMDC project and prototype, which is a client/server, Web-based audio- and data-conferencing system. Our approach to this highly active research and business area is to develop a system that is easy to operate, scalable, and universally available, i.e., platform-independent. Consequently, the MMDC client code is written in Java, in order to allow the same code to run under different operating systems. Also, a conference participant is not required to have any special hardware beyond a modem or LAN adapter. The client software uses the HTTP standard Internet protocol to communicate with the MMDC server, thus allowing global access and collaboration, even through firewalls.

For making conferencing reservations and changing such reservations, MMDC allows clients to use the Internet to access phone directories. MMDC allows conference chairpersons to push Web-page content to clients very efficiently, in the form of preloaded HTML slides or information just retrieved from the Web, in a way that guarantees a consistent Web-page view for all of the conference participants. In MMDC, we have integrated audio control facilities that allow the establishment, and control, of high-quality voice conferencing among the conference participants.

The MMDC project, funded by the IBM First-of-a-Kind (FOAK) program, has been developed with the audio- and data-conferencing market as its target, yet MMDC opens up a wide range of application opportunities, including education, telecommuting, remote shopping, enhanced chat rooms, town meetings, Customer Care, and electronic auction events.

The first prototype of the conferencing application was tested in a field trial with a North American telephone company in the spring of 1997. This field trial involved ordinary Internet services subscribers and operators of audio-teleconferencing services. The ability to use a Web browser was the only requirement for someone to use MMDC. This established the fact that our prototype was easy to use. Subsequent to the field trial, enhancements have been made to the platform-independent client and the performance and functionality of the conference control and the Web processor. The prototype server runs on the AIX operating system, and the prototype client requires the Netscape** 4.0x browser. Subtle differences between the way Netscape and other browsers handle JavaScript** limit the ability of this version of the system to be truly platform-independent, but creation of modified versions of the client code for other browsers is being considered.

The MMDC prototype is currently being demonstrated to IBM's customers in the Industry Solution Laboratory (ISL) in Hawthorne, New York. This technology is being transferred to the Telecom and Media Industry Solution Unit (T&M ISU) development division in LaGaude, France. We are targeting teleconferencing service providers for running this service with the use of their automated audio-bridging infrastructures and offering it first as a service within IBM, and then to external customers. Several parts of MMDC teleconferencing technology are now being ported to Lotus Notes** clients, with a goal to automate the procedure of audio-conference reservation and initiation, and to enhance the audio-teleconferencing interfaces with a visual interface.

Acknowledgments

The authors are deeply indebted to their colleagues Colin G. Harrison, Jurij R. Paraszczak, Edith H. Stern, and Barry E. Willner for the ideas, leadership, and support they so generously provided to the MMDC project. The authors are also thankful for the full support and encouragement of the IBM Corporation and various IBM divisions. The careful reading and fruitful comments of the reviewers and editors greatly improved the presentation of this paper.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., The Open Group, Netscape Communications Corporation, Microsoft Corporation, AT&T, Intel Corporation, Apple Computer, Inc., DataBeam Corporation, or Lotus Development Corporation.

References

1. B. Aldred, *Desktop Conferencing: A Complete Guide to Its Applications and Technology*, McGraw-Hill Book Co., Ltd., London, 1996.
2. V. Bush, "As We May Think," *The Atlantic Monthly*, July 1945 (a recent reproduction is available from *http://www.isg.sfu.ca/~duchier/misc/vbush/vbush.shtml*).
3. D. Engelbart and W. English, "A Research Center for Augmenting Human Intellect," *Proceedings of the Fall Joint Computing Conference*, Montvale, NY, 1968, pp. 395–410.
4. S. Greenberg, *Computer Supported Cooperative and Groupware*, Academic Press, Ltd., London, 1991.
5. R. Rada, *Groupware and Authoring*, Academic Press, Ltd., London, 1996.
6. K. Schmidt, Ed., *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, Kluwer Academic Publishers, Inc., 1992–1998.
7. C. Blum and R. Molva, "A Software Platform for Distributed Multimedia Applications," *Proceedings of the International Workshop on Multimedia Software Development*, Berlin, Germany, March 1996.
8. M. Arango, L. Bahler, P. Bates, M. Cochinwala, D. Cohrs, R. Fish, G. Gopal, N. Griffeth, G. E. Herman, T. Hickey, K. C. Lee, W. E. Leland, C. Lowery, V. Mak, J. Patterson, L. Ruston, M. Segal, R. C. Sekar, M. P. Vecchi, A. Weinrib, and S.-Y. Wuu, "Touring Machine System," *Commun. ACM* **36,** No. 1, 68–77 (1993).
9. R. Strom, G. Banavar, K. Miller, A. Prakash, and M. Ward, "Concurrency Control and View Notification Algorithms for Collaborative Replicated Objects," *Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS '97)*, Baltimore, MD, May 1997, 194–203.
10. A. Prakash and H. S. Shim, "DistView: Support for Building Efficient Collaborative Applications Using Replicated Objects," *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, Chapel Hill, NC, October 1994, pp. 153–164.
11. J. H. Lee, A. Prakash, T. Jaeger, and G. Wu, "Supporting Multi-User, Multi-Applet Workspaces in CBE," *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'96)*, Boston, MA, November 1996, pp. 344–353.
12. K. Maly, H. Abdel-Wahab, M. C. Overstreet, C. J. Wild, A. K. Gupta, A. Youssef, E. Stoica, and E. S. Al-Shaer, "Interactive Distance Learning over Intranets," *IEEE Internet Computing* **1,** 60–71 (1997).
13. M. Roseman and S. Greenberg, "Building Real Time Groupware with GroupKit, a Groupware Toolkit," *ACM Trans. Computer–Human Interaction* **3,** No. 1, 66–106 (March 1966).
14. B. K. Aldred, G. W. Bonsall, H. S. Lambert, and H. D. Mitchell, "An Architecture for Multimedia Communication and Real-Time Collaboration," *IBM Syst. J.* **34,** No. 3, 519–543 (1995).
15. T. J. Frivold, R. E. Lang, and M. W. Fong, "Extending WWW for Synchronous Collaboration," *Computer Networks & ISDN Syst.* **28,** 69–75 (1995).
16. S. Greenberg, "Collaborative Interfaces for the Web," *Human Factors and Web Development*, C. Forsythe and E. Grose, Eds., Lawrence Erlbaum Assoc. Press, Hillsdale, NJ, 1997, pp. 241–254.
17. M. B. Twidale, D. M. Nichols, and C. D. Paice, "Browsing Is a Collaborative Process," *Info. Proc. & Manage.* **33,** No. 6, 761–783 (1997).
18. J. Zinkl, K. Maly, and M. Zubair, "A Coordinated Browsing System," *Technical Report TR-97-29*, Old Dominion University, Norfolk, VA, May 1997.
19. J. Hill, L. Ozer, and T. Mace, special report on Internet collaboration, *PC Magazine*, pp. 103–239 (October 1996).

20. *Teleconferencing Markets & Strategies: The Convergence of Price, Performance & Plumbing*, Forward Concepts, Tempe, AZ, 1995.
21. W. Stallings, "The Backbone of the Web," *Byte*, pp. 63–64 (October 1996).
22. J. December and M. Ginsburg, *HTML and CGI Unleashed*, Samsinet Publishing Co., Indianapolis, 1995.
23. V. W. Mak, "Session Management for Distributed Multimedia Applications," *Proceedings of MULTIMEDIA '94*, Kyoto, Japan, May 1994, pp. 6/2/1–6/2/5.
24. *Enhanced Services: MSP System Architecture*, TSC Architecture Group, IBM LaGaude, France, 1996.
25. V. Jacobson and S. McCanne, "The LBL Audio Tool vat," Manual page, *ftp://ftp.ee.lbl.gov/conferencing/vat/*, July 1992.
26. H. Schulzrinne, "Voice Communication Across the Internet: A Network Voice Terminal," *Technical Report TR 92-50*, Department of Computer Science, University of Massachusetts, Amherst, July 1992.
27. H. Eriksson, "MBone: The Multicast Backbone," *Commun. ACM* **37,** No. 8, 54–60 (1994).
28. R. J. Vetter and C. Jonalagada, "A Multimedia System for Asynchronous Collaboration Using the Multicast Backbone and the World Wide Web," *Proceedings of the First IEEE Annual Conference on Emerging Technologies and Applications in Communications*, Portland, OR, May 1996, pp. 60–63.

[Apple] Apple's QuickTime Conferencing, *http://devworld.apple.com/dev/techsupport/develop/issue26/blackletter.html.*
[ATT] AT&T's On-Line Teleconference Service, *http://www.att.com/teleconference/online.html.*
[CP] IBM's CallPath and DirectTalk, *http://www.networking.ibm.com/cti.*
[CT] Netscape Conference and CoolTalk Meeting Room, *http://www.q5.com.*
[DataBeam] Databeam's neT.120 Conference Server, *http://www.databeam.com/net120.*
[HTML] HTML Reference Specification, W3C Recommendation, *http://www.w3c.org/MarkUp.*
[HTTP] HyperText Transfer Protocol, *http://www.w3c.org/Protocols.*
[Intel] Intel's Proshare, *http://www.intel.com/proshare.*
[Java] Java Language, *http://java.sun.com/products.*
[Microsoft] Microsoft's NetMeeting, *http://www.microsoft.com/netmeeting.*
[Sprint] Sprint's Internet Conference Center, *http://www.sprintconf.com.*
[T.120] T.120 Real Time Data Conferencing ITU Standard, *http://www.imtc.org/i/standard/itu/i_t120.htm.*
[WW] AT&T's WorldWorx Services, *http://www.att.com/worldworx.*

Received October 23, 1996; accepted for publication August 26, 1997

Chatschik Bisdikian *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (bisdik@us.ibm.com).* Dr. Bisdikian received the Diploma in electrical engineering from the Polytechnic School of the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1983 and the M.Sc. and Ph.D. degrees, both in electrical engineering, from the University of Connecticut, Storrs, in 1985 and 1988, respectively. In 1989 he joined the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, as a Research Staff Member. His current research interests are mobile communication networks, home networking, multimedia and broadband communications, and modeling and performance evaluation of high-speed networks. He has published more than 60 technical papers, and he holds a patent. Dr. Bisdikian was a 1995 finalist for the Eta Kappa Nu Outstanding Young Electrical Engineer Award. He served on the Editorial Board of the *Telecommunication Systems Journal* from 1993 to 1996. He is a voting member of the IEEE 802.14 standards committee, a senior member of the IEEE, a member of the New York Academy of Sciences, and a lifelong member of the Eta Kappa Nu and Phi Kappa Phi honor societies. He is included in the 1998–1999 edition of *Who's Who in Science and Engineering.*

Stephen Brady *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (sxb@us.ibm.com).* Dr. Brady holds a Ph.D. in mathematical logic from Cornell University. He joined IBM in 1982 as a member of the Advanced C3 organization of the Federal Systems Division in Owego, New York. In 1984 he moved to the Thomas J. Watson Research Center in Yorktown Heights, New York, to work on problems in the area of network management. Since he joined the Research Division, Dr. Brady's work has included contributions in the use of expert systems for network management, definition of management architectures for IBM network products, and applications of intelligent agent technology. Dr. Brady is currently a member of the Network Interactivity group, conceiving and developing multimedia solutions for the telecommunications industry.

Yurdaer N. Doganata *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (yurdaer@us.ibm.com).* Dr. Doganata received his B.S. and M.S. degrees from the Middle East Technical University, Ankara, Turkey, in 1981 and 1983, respectively, and the Ph.D. degree from the California Institute of Technology in 1987, all in electrical engineering. From 1987 to 1989, he was a postdoctoral Research Fellow at the California Institute of Technology and a lecturer at California State University, Long Beach, working on queueing models of mobile satellite-communication systems and integrated networks. In 1989, he joined the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, as a Research Staff Member, working on the performance modeling and evaluation of various computer and communication systems. He worked in several areas, including high-speed switching systems, multimedia servers, and intelligent transportation systems. He has published a number of articles in books, journals, and international conferences. His current interests are multimedia collaborative applications and Internet technologies.

Davis A. Foulger *IBM Global Services, East Fishkill facility, Route 52, Hopewell Junction, New York 12533 (folger@us.ibm.com).* Dr. Foulger received a Ph.D. in human communication from Temple University; he has spent much of his IBM career creating and researching new means of computer-mediated human interaction. Currently the overall architect of the IBM Global Business Partner Initiatives, he also manages IBM's forum and package repository strategy for the IBM CIO's office. His primary research interest is "the process of media," a theory that explores both the ways in which people create and continuously reshape communication media to solve real problems and the factors that may predict a new medium's prospects for success or failure (*http://pages.prodigy.com/mediaspace*). He also has current research interests in agent-mediated human-communication systems and context-oriented programming systems. He has authored both widely influential software (including the first widely used tag-based hypermedia system and the first direct-manipulation subdirectory tree manager), and a variety of articles, conference papers, book chapters, and patents (including an invited paper on rights, responsibilities, and Internet-based computer media for the National Science Foundation. Dr. Foulger has received an IBM Technical Achievement Award and an IBM Golden Circle Award.

Franco Marconcini *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (francom@us.ibm.com).* Mr. Marconcini received the B.S. degree in computer science from the University of Milan, Italy, in 1990 and the M.Sc. degree in electrical engineering from Columbia University, New York, in 1996. After graduation, he worked as a software engineer, designing and developing distributed information systems for commercial applications. In 1993, he was awarded a Postgraduate Fellowship from the University of Milan and joined the COMET Group at the Center for Telecommunication Research at Columbia University as a Visiting Scholar. In the COMET Group, he played a major role in the design, development, and deployment of the ATM network services of the xbind project. In 1996, he joined the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, working on interactive multimedia services for e-business applications. His current focus is on thin client architectures that use Web and Java technologies to allow for global and ubiquitous services.

Magda Mourad *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (magdam@us.ibm.com).* Dr. Mourad is currently a Research Staff Member and Manager of the Network Interactivity Department at the IBM Thomas J. Watson Research Center. Her current research interests include multimedia collaboration tools and applications, Web-based enhancements to telecommunications services, and multimedia communications over the Internet as well as emerging broadband networks. She received her Ph.D. in computer engineering from the École des Mines de Nancy, France, in 1982 in multimedia document presentation and transfer protocols. She received a B.Eng. in electrical engineering and an M.Sc. in computer engineering from Alexandria University, Egypt, in 1974 and 1977, respectively. Dr. Mourad has received an IBM Outstanding Technical Innovation Award, a Research Division Award, and a Research Division Technical Group Award. She is a senior member of the IEEE.

Howard L. Operowsky *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (howardo@us.ibm.com).* Dr. Operowsky is a Senior Software Engineer in the Network Interactivity Department at the IBM Thomas J. Watson Research Center. He is currently interested in the design of collaborative multimedia applications. Before joining the Research Division in 1995, he worked in the areas of MVS performance analysis, the design and development of compilers and programming languages for high-end parallel systems, and architecture of the AIX/ESA operating system. In 1989, Dr. Operowsky received a Ph.D. in computer science from New York University while participating in the IBM Resident Study program. In 1992, he received an IBM Outstanding Technical Achievement Award for the AIX/ESA program management architecture. He is a member of the IEEE.

Giovanni Pacifici *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (giovanni@us.ibm.com).* Dr. Pacifici received the *Laurea* in electrical engineering and the *Research Doctorate* in information science and telecommunications from the University of Rome *La Sapienza* in 1984 and 1989, respectively. His research activities were focused on the design and performance evaluation of access control protocols for local and metropolitan area networks. In the course of his doctorate studies, he visited the Center for Telecommunications Research at Columbia University, where he designed and implemented a monitoring and multimedia traffic generation system for MAGNET II, a high-speed multimedia network. In 1989, he joined the Center for Telecommunications Research as a Research Scientist, where he led several research activities that focused on the design and evaluation of real-time control and monitoring systems for high-speed multimedia networks with quality-of-service guarantees. He was also involved in the design of management architectures for broadband networks as well as their evaluation by use of high-performance parallel computers. In 1995 he joined the IBM Thomas J. Watson Research Center, where he currently works in the areas of multimedia services and service-management architectures. Dr. Pacifici is a senior member of the IEEE and a member of the ACM.

Asser N. Tantawi *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (tantawi@us.ibm.com).* Dr. Tantawi received the B.S. and M.S. degrees from Alexandria University, Egypt, and the Ph.D. degree from Rutgers University in 1975, 1978, and 1982, respectively, all in computer science. He joined the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, in 1982, as a Research Staff Member in the Computer Sciences Department. Dr. Tantawi has published numerous articles in books, journals, and international conferences. His fields of interest include telecommunications and information-network applications, multimedia systems, mobile computing, high-speed networking, and performance modeling and analysis. He received an IBM Outstanding Technical Innovation Award and two IBM Plateau Invention Achievement Awards. Dr. Tantawi is a senior member of the IEEE, a member of the ACM, INFORMS, and IFIP WG 7.3 on Computer Systems Modeling. He served as an ACM national lecturer and as a guest editor for *Performance Evaluation*.