RC 20697 (91740) 1/22/1997
Computer Science/Mathematics 22 pages

IBM Research Report

MultiMedia Digital Community:
A web-enabled multimedia collaboration system

Chatschik Bisdikian, Stephen Brady, Yurdaer N. Doganata, Davis Foulger
Franco Marconcini, Magda Mourad, Howard L. Operowsky
Giovanni Pacifici, and Asser N. Tantawi

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, New York 10598

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

MultiMedia Digital Community: A web-enabled multimedia collaboration system

Chatschik Bisdikian, Stephen Brady, Yurdaer N. Doganata
Davis Foulger, Franco Marconcini, Magda Mourad
Howard L. Operowsky, Giovanni Pacifici, Asser N. Tantawi

IBM Research Division
Thomas J. Watson Research Center
30 Saw Mill River Road, M/S H2-C24
Hawthorne, NY 10532

e-mail: {bisdik, sxb, yurdaer, folger, mfranco, magda, howardo, giovanni, tantawi}
@watson.ibm.com

Abstract: *The convergence of telecommunications and computers is expected to bring a real revolution in 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 to browse and navigate through the Internet, new easy-to-use tools are being developed to simplify and enhance the on-line interaction of people.*

In this paper, we will describe the multimedia digital community (MMDC) project and prototype, which is a web-enhanced teleconferencing solution. The MMDC solution is an open system, that is developed on the client-server model, and allows non-homogenous clients to use the Internet to establish and participate in audio and/or data conferences using a variety of audio and computer systems.

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 is only one of a myriad of traffic types traveling the Internet, with graphics, images, video and audio clips consuming the bulk of the networking resources on the Internet.

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, e.g., T3 lines from MCI are already in use. New communications protocols are also being developed that would allow certain types of traffic to be delivered to their 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, in the last few years, the rapid development of the *World-Wide Web* (WWW, or just "web") browser technology, which provides an easy *point-and-click* approach for navigating through the tons of multimedia information scattering on millions of Internet-connected computers around the globe, has brought the riches of Internet to the desktops of most people with no more than a modem equipped personal computer.

With the increasing amount of information available on-line, people can access and manipulate it remotely, thus, for example, enabling employees to telecommute. Similarly, residential customers can use on-line services for things like remote shopping, remote education, theater seat reservations, etc. In contrast with these one-on-one interactions between a client and the information accessed, several business transactions involve some sort of multi-party collaboration and interaction. Hence, multimedia versions of office environments and meeting-rooms need to be developed on desktop computers that would allow people to collaborate remotely in virtual office surroundings existing only on their desktops and to meet and conduct business with their colleagues, peers, and business associates on-line.

The idea of integrating multimedia and collaboration is not a new one. Indeed, Vannavar Bush's visionary description of the *MEMEX* in 1946, a networked terminal-like device, combined multimedia and collaboration. Yet, despite their long history and their well regarded benefits and great market interest [3], multimedia collaboration systems have been slow to develop into practical and broadly available media. Indeed, it can be argued that the only widely used multimedia collaboration platform in use today is a combination of audio conferencing and fax in which pages and presentations are faxed between meeting participants in advance.

The problem in achieving practical computer-based multimedia collaboration systems, we believe, has been a lack of standards for non-voice multimedia content delivery and interaction. However with the growing usage of the *hyper-text markup language* (HTML) in preparing and linking information on the web, this problem can now be addressed. Taking advantage of the emerging HTML-based web hyperdocument publication standards provides a practical base for building standard and broadly available multimedia collaboration solutions.

Realizing that a standards-based, hence, open and widely accepted, web-enabled conferencing solution could be possible, the idea of a *MultiMedia Digital Community* (MMDC) was conceived with the objective of marrying the desire for on-line interaction and collaboration, using text, graphics, and voice communications, with the user-friendliness and pervasiveness of web-based multimedia browsers interfaces. The MMDC is a *client/server* collaborative solution that has been guided from its conception to its realization as a working prototype by the need to develop a system that is *open standards-oriented*, *platform independent* with *low barriers of use* on the client side, and easily *migratable* onto different platforms and *scalable* on the server side.

In the rest of this paper, we will present the MMDC system from the grounds up. The organization of the paper is as follows. In section 2, we present a brief history and a description the MMDC system and its objectives, while in section 3, we provide an review of the MMDC functions. In section 4, we present the architecture of the MMDC and elaborate on on the key software modules so far developed for our working prototype of MMDC. In section 5, we compare MMDC with other audio-/data-conferencing solutions and expand on its applications range. Finally we conclude in section 6, with a summary of our work.

2 The MMDC project and system

The MMDC project started out as an exploration of real-time multimedia, multi-party 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 a standard form of multimedia content was being achieved. Moreover, we were interested in creating a solution that would be of interest to the wireline and cable businesses supported by the IBM Telecommunications and Media Industry Solutions Unit (T&M ISU).

Therefore, our goal was not merely to create an application where individual Internet users could collaborate in peer-to-peer relationships, much as other products now available were attempting [6], but to provide an infrastructure by which existing businesses could legitimately exploit the growing power and popularity of the Internet, and the intranets,[1] communications to enhance their current business products. A *client/server* architecture for the MMDC end-to-end solution serves this need much better than a peer-to-peer architecture. For example, in our first target application, the integration and enhancement of traditional audio-conferencing with document-conferencing, our goal was not to simply 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

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

MMDC server could support and sell enhanced conferencing services to their customers, the conference *participants*.

The selection of our first target application has been 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 a rate of about 33% for, at least, the remaining of this century [3]. Additional application areas for MMDC include distributed multimedia classrooms, town hall-type meetings Internet social shopping groups, audio on-line services, etc.

Given this perspective, in addition to the adoption of a client/server architecture, we quickly arrived at several design points for MMDC.

Heterogeneous environments: The solution had to support several forms of heterogeneous environments. Most basically, everyone has a telephone. Participants will naturally expect that if they have access to the Internet from their desks, alongside their telephones, any service sponsored by the TelCo will be as universal as telephone service itself. Thus, we had to support a heterogeneous set of participants within a single conference. Participants should no more have to worry about each other's Internet platforms before scheduling a collaborative teleconference than they worry about each others phone equipment or long distance carrier when scheduling a conference call today. In addition, we did not want to belabor a conference service provider, e.g., a TelCo, with supporting various levels of platform-specific binaries. Thus, the decision was made that all code destined for execution on a participant's platform be written in the Java object-oriented language.

Also, there were no predefined dependencies on the primary server platform. Although the initial implementation of the MMDC Server was developed on IBM's AIX operating system, there are no inherent system dependencies.

Application flexibility: We knew that collaborative teleconferencing was an obvious application choice. Even the later enhancements we contemplated, such as conference transcription services, and voice-based indexing are relatively routine applications. Also, the MMDC system had to grow along with the Internet. Thus, the underlying architecture had to allow easy extensions. The system architecture had to permit applications to replace each other or interact with each other during the course of a conference depending on the type of conference.

Scalability: The architecture had to scale the expected growth in Internet usage and function. We expect, of course, that we would easily obtain some scalability by off-loading as much function as possible onto the the participants' computers. But we also know that our server must scale as well and grow into a heterogeneous network itself as individual MMDC customer, e.g., the TelCos, conditions warrant.

Participant ease of use: As indicated above, our ultimate goal for MMDC is that it must be as simple to use as the telephone. To this end, most interaction with the MMDC server is within the context of the user's own browser. Ease of use also extends to installation and configuration of participant software. Our original design point was that participants would not have to preload or install any MMDC code prior to a conference. The Java operating environment permit us so by dynamically downloading Java applets that can then run locally within the boundaries of a Java-capable browser. In addition, the client code requires no configuration.

Figure 1 shows an MMDC system where MMDC service customers at business or home locations connect to the MMDC service location. There are two logical paths in this connection. One path is for the voice communications and one is for the data communications. Currently, the voice connection to MMDC is through public-switched telephone network (PSTN), although, other network are under consideration as well. For the data communications, the logical path is the Internet to which a customer may be connected directly via his office local area network (LAN) attachement, or indirectly via a modem connection to an Internet service provider.

The MMDC server complex provides all the necessary facilities for achieving the audio and data conferencing. Note that voice only participants may be allowed to access some of the MMDC functionalities, e.g., MMDC may call and admit a person into a conference who only has a telephone set. The elements of the MMDC complex will be discussed later in the paper.

Figure 1 depicts an MMDC with "independent" voice and data networks to and from the MMDC center. This is not a requirement though on MMDC, which can operate over a single network, with both voice and data sharing either the PSTN or the Internet. Yet, the dual-line solution in figure 1 is the result of input from customers, mainly telecommunications companies (TelCos), and feedback from actual users of MMDC who prefer the suitability of PSTNs to deliver high-quality voice communications. 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 mostly likely be used first, the dual-line 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 at an office. The two-line MMDC solution is part of the current FOAK realization of the MMDC solution.

3 The MMDC system functions

The voice and data-conferencing integration in MMDC provides an effective alternative to the type of teleconferences typically held today, where conference materials are faxed to all conference participants prior to the conference. Yet, 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 wanted to eliminate the need for delivering any presentation materials to conference participants prior to the conference, and

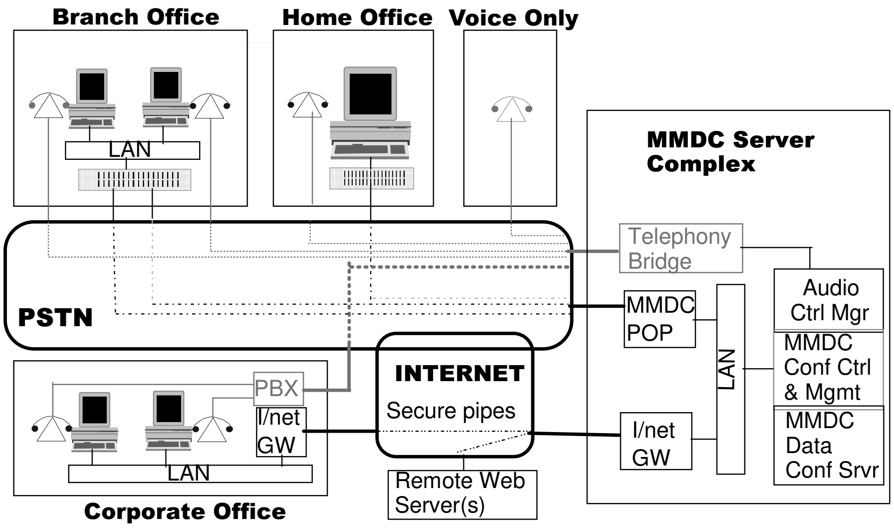

Figure 1: An MMDC enabled system.

simplify the process of distributing additional material during the conference, should the need arise. All interactions will occur both through text and graphic data files over the web in practically real-time, but also through actual voice-communications.

A client, the MMDC user, can access all MMDC functions by first opening the *uniform resource locator* (URL) for the MMDC home page and identifying himself. After that, the user is presented with a menu of all MMDC functions. In analogy to the fax-enhanced audio-conferencing, in MMDC, we allow clients, the conference *participants*, to access the MMDC service over the Internet in order to reserve time-slots for conferences, change conference reservations, select and join an on-going conference, share and comment on one's slide-presentation, review the participant's list, review a conference transcript, or previously shown slides, etc. The MMDC server, at a conference participant's request, will establish voice-conferencing with all or some of the members of a conference by dialing out the participants' phone number. Note that MMDC allows voice-only conference participants, if another conference participant supplies the phone number of the voice-only participant to MMDC. A participant could further control the audio portion of the conference by clicking on an audio, mute, volume, etc, buttons on the conference web-page.

For each conference, a particular participant is identified as a conference *chairperson* that moderates the whole conference, and gives the floor to anyone requesting it. He will be able to preview and edit slide presentations, prior to broadcasting them to the conference participants. He will be able to "push" slides (preloaded data or just retrieved from anywhere on the web) to all the participants, thus assuring that all conference participants have a common view and perception of the current state of the conference. As the circumstances dictate, the conference chairperson will be able to dynamically disable or enable the addition of new conference participants, or dynamically disable or enable audio or data communication with conference members.

The key objective of any conference meeting is the creation, presentation, and delivery of information and content to be shared among the conference participants. In MMDC, this is accomplished as follows.

Creation of a presentation: A conference member is able to load his presentation through the upload MMDC panel, by identifying the necessary files on his local computer. Currently, we require the presentation to be written in the HTML. We expect to support all major presentation formats, at least through the use of filters which convert them in HTML files. After the presentation "slides" are accepted by the system, the chairperson is able to preview them, and make any necessary changes. The chairperson may continue to review and revise a presentation even *during* the conference.

Delivery of the presentation: We call the collection of presentation files a *slide-tray*. Slide-trays are created independently of any conference. The chairperson gains access to a particular slide-tray by asking for it when he joins the conference. The chairperson is then able to view the the list of slides by clicking on a button in his window. Then, selecting a file in the list causes it to be broadcasted to all participants connected via the web.

We chose HTML as our presentation format to allow the chairperson the full capability of the web. If any presentation contains a URL reference, the chairperson can cause everyone to visit that site by simply clicking on the reference, an activity we call *group surfing*. We also allow the chairperson to start group surfing by specifying an arbitrary URL during the conference. The chairperson can of course choose to display a page from his slide-tray at any time.

In support of the above functions, we have identified five key elements that have to be developed for MMDC. These elements will be further discussed in the next session.

1. A *registration desk*, which takes on much of the functions associated with a human operator in a current audio teleconferencing systems, e.g., it will authenticate clients, allow meetings to be scheduled, etc. The registration desk service, to be called the *concierge*, will be first point of contact of a client with MMDC, and it will accessible via a web-browser.

2. A *floor control* capability for initially allowing a moderator to control who can talk and select content for display, and eventually for allowing such control to be done automatically on the basis of collaboration rules.

3. A *web-projector* for broadcasting collaborative content, called the *slides*, into connected web-browsers. This feature will eliminate the need for faxing conference/presentation material among the conference participants.

4. A *web-page processor* for retrieving web-material that are referenced in slides via URL addresses.

5. A *request processor architecture* that will allow for collaborative applications, including whiteboards, shared file cabinets, and other collaborative services, to be attached to the MMDC server as needed.

Almost all of these functions are located in a central server. Client function is, with the exception of a small Java application located on the client workstation, handled entirely through a standard Java-enabled web-browser and existing telephone and/or Internet-based audio services. It is expected that this approach will make multimedia collaboration practical and broadly available.

4 The MMDC software architecture

MMDC is a distributed application with code executing on each participant's workstation, as well as within the MMDC server complex. One of the key architectural ideas of MMDC is that its function should be universally available. Thus, given the general acceptance of HTML files as the common means of communication on the web, we decided, as stated before, to adopt HTML as the basis for content preparation.

One concern with this choice was the stateless nature of the *hyper-text transfer protocol* (HTTP) [4, 8] used for the transport of HTML-coded content over the web. Every time that 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 requester may need to access data on that server. Thus, the HTTP protocol poses several problems for a collaborative system such as ours. One is the obvious overhead in retrieving data from the server. In a conference setting, communication will be within a single server and multiple clients. Thus a more efficient approach would be to maintain these connections for the duration of the conference. A second concern was that for a conference to proceed smoothly, data must be delivered to the clients simultaneously. In the standard HTTP protocol, where each client needs to open a connection with the server, requests from a conference will come in bursts thus allowing the possibility that some client requests will be refused, causing a delay in that client receiving the next page, and ultimately delaying the conference. Clearly, the larger the conference, the greater the likelihood of this occurrence.

In the rest of this section, we will present the key software elements of the MMDC server and client.

4.1 The MMDC server

The MMDC server is divided into various components in order to scale according to the application and environment. Figure 2 shows the software modules that comprise the MMDC server. Several of these modules may be used off-the-shelf and can be added as needed, like a transcription manager, accounting, etc. The core of the MMDC server development though focuses on the following four modules: *concierge*, *conference control*, *web-page processor*, and *audio control manager*. At a very high level, the web-page processor and the audio control manager together with the *conference control* make up a single basic conference. Yet, their features are sufficiently distinct to merit separate development. Figure 2 reflects this lower level description of conferences, where we explicitly show the web-page processor, while the audio control manager is imbedded within the *connection objects* module. Finally, recall that the request processors, which can also be considered part of *conference control*, are attached to MMDC server as needed to add enhanced collaborative applications.

4.1.1 The concierge

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

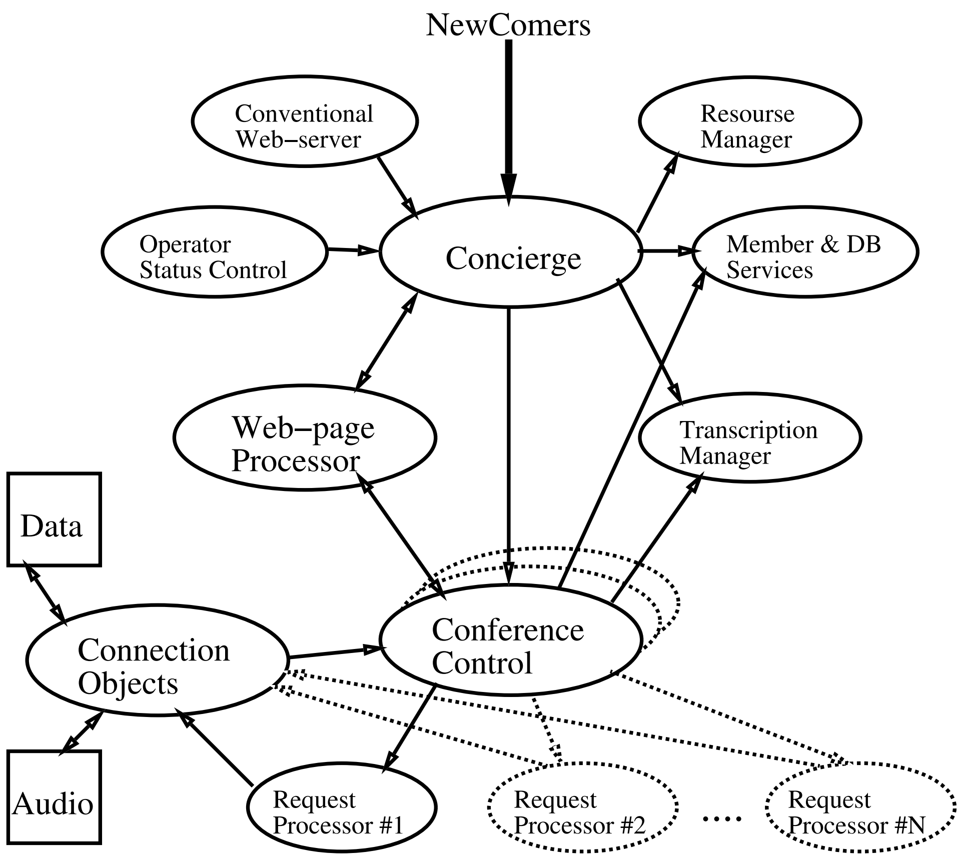

Figure 2: The MMDC server software architecture

enterprise billing records.

The *concierge* interacts with clients via any standard web-browser and web-server, using *common gateway interface* (CGI) scripts [9]. When a conference is scheduled to begin, the *concierge* "creates" that conference by creating an instance of *conference control*. When a client joins a particular conference, his first contact is with the *concierge* which performs an initial security check, and then orchestrates a hand-off which results in the cancellation of the client-*concierge* connection and the creation of a persistent connection between the client and the *conference control* component managing the requested conference. All further client data interaction with the server will be with *conference control*.

4.1.2 The conference control

While the *concierge* "welcomes" customers to MMDC, an actual conference takes place and is managed under the auspices of *conference control*. The *conference control* module plays two key roles in delivering the type of collaborative applications envisioned for MMDC. First, it is this component which joins together participants into collaborative groups, and maintains the status of communications 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 positive indicators 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 this approach.

1. *Scalability*: By managing each instance separately, and eliminating platform dependencies within conference control, the MMDC server can scale with demand simply by adding additional processors. In addition, if telephone switching hardware is distributed, *conference control* can be placed on the processor which can communicate with the telephone hardware most efficiently.

2. *Security*: By having each conference managed by a separate entity, we can guarantee the security of data within each conference. Participant clients deal only with a single *conference control* instance, and no *conference control* instances ever communicate with each other.

3. *Implementation*: By focusing on managing only one conference, the *conference control* function is easier to implement.

We envision *conference control* to take on different forms as different collaborative applications are created. But based on our experiences in creating the initial conferencing application, we believe that all will maintain a similar general structure. The current *conference control* component is comprised of:

- a *communications* component: All accesses to data outside the MMDC environment are handled by a single communications component which provides a general abstract interface to the rest of MMDC. The primary reason for this approach was to eliminate platform dependencies. For example, *conference control* currently executes only on AIX platforms using the Internet TCP/IP suite of communications protocols, but it could be easily ported by simply changing the communications component to use, for example, the Winsock interface instead. None of the internal *conference control* function would have to be changed. Similarly, the Communications component hides the precise nature of our telephony hardware, so MMDC can support new hardware by changing only the audio layer within the Communications component.

  I/O requests handled by the Communications 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 connection and audio connection are made and remain intact throughout the conference. Similarly, at the end of the conference, or when the participant decides to leave, the membership component is responsible for recovering resources.

- several *request processor* components: Our intent was to develop a conference manager which would be easily configurable to meet the different requirements of various collaborative activities, services, and resources that users wish to include in their online collaborative environment, e.g., a shared whiteboard, access to directory services, etc. We achieve this with the addition of request processors, which is our name for these individual functional units. *Conference control* defines an object class for request processors to allow us to dynamically install request processors during a conference, as well as access request processors running on different platforms - further supporting platform independence and scalability.

4.1.3 The web-page processor

At a minimum, an MMDC data-conference offers the capability of web-page broadcast to all the conference participants. As indicated previously, we decided initially that all conference content would be coded in HTML and trasmitted to and from the conference participants using HTTP. In order for us to address the HTTP performance concerns previously mentioned, and the scalability of our MMDC solution, we decided to support a *server push* model rather than the standard *client pull* model. In addition, we wanted to allow conference participants to have the full power of the web. For example, we wanted to allow *group surfing* whereby the entire conference could surf the web together via a single click on a URL reference, by a single conference participant, just as an individual surfs the web. We also intend to allow individual participants to branch off and take separate trips through the web. Furthermore, we wanted to provide these capabilities within the context of an individual's own browser, and without change to any of the browser's options or configuration. The *web-page processor* (WPP) is the basis for this function within MMDC.

Before any HTML page is viewed by any conference participant, it is obtained, analyzed and modified by the WPP. The WPP currently has four basic responsibilities:

1. Obtain requested URLs, and all imbedded files: During a conference, the WPP becomes the web proxy for each conference participant, and retrieves requested HTML pages. In addition, the WPP will obtain any other files referenced within the HTML page retrieved, including image files and Java applets. This reduces the likelihood of delay due to contention on the real web server.

2. Locate and modify all HTML anchors and file references: This is the means by which we guarantee that *conference control* is notified when any conference participant ventures off into the web. After WPP obtains an HTML page, it transforms each URL reference into an MMDC-specific command which causes the MMDC client software to gain control and transfer the request to the WPP function within *conference control*.

3. Transfer all files obtained to the conference participants: After retrieving the data from the web, the WPP forwards the data to the individual participants, along with control information used by the MMDC client software to manage the data and deliver it to the participant's browser upon request.

4. Process user-generated slides: The primary function of a conferencing system is to allow the participants to share their ideas. Prior to the conference, users preload their presentation slides onto the MMDC server through a function provided by the *concierge*. WPP processes each slide in much the same way that it processes pages from the web so that conference slides have all the "hyper-link" functionality available, and so that web pages may be displayed in a seamless fashion during the conference.

4.1.4 The MMDC audio control manager

While coordination of the various communications channels with conference participants is controlled at an abstract level through a software representation of each participant, in the current implementation, it is possible for *conference control* to manage only the data channels directly. To create and manipulate the audio domain, *conference control* invokes the services of an *audio control manager* (ACM). The latter software masks *conference control*

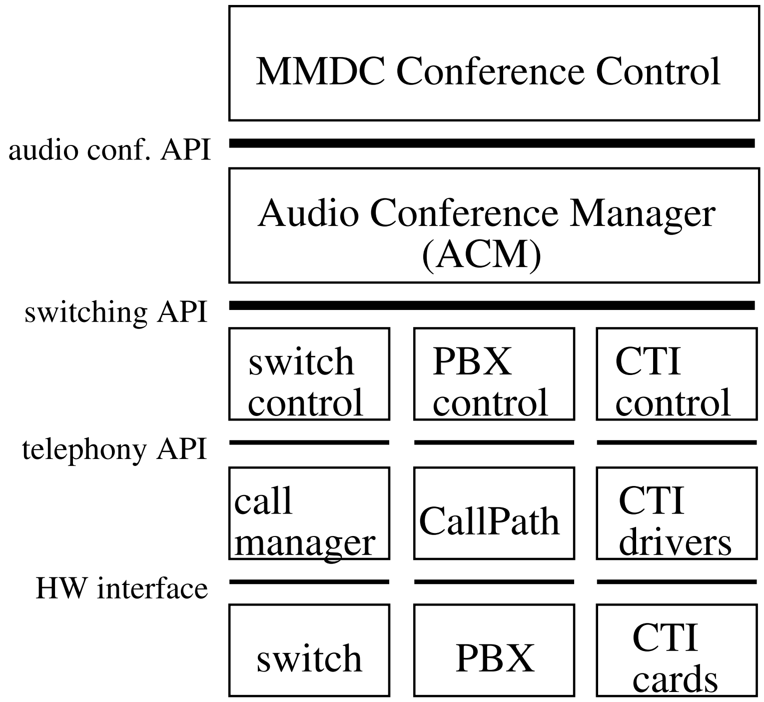

Figure 3: The MMDC ACM architecture

from the details of the audio implementation, helping realize some of the network technology independence that was a goal throughout the system design.

Various audio-conferencing platforms were considered for MMDC, some of them have actually been tested we well.

IP conferencing: Utilize existing Internet-based audio-conferencing tools. This approach lacks good quality for voice communications, as confirmed from our tests, while does not scale gracefully, since there exists no widely acceptable voice-coding schemes, and the limited capabilities of personal computers in mixing and handling multiple voice sources.

PSTN conferencing: Audio conferencing may be realized "in" the PSTN, using the conferencing services provided by TelCos. ISDN signaling supports conferencing operations. For non-ISDN systems, one may use the signaling protocols provided by the carriers. Note that there is no guarantee that an MMDC solution buyer will provide a audio-conferencing service of his own. Thus, our current efforts focus on the following three alternatives, which are also reflected in figure 3.

Open-switch conferencing: Using an open-architecture switch with appropriate digital conferencing cards provides a lot of flexibility in terms of switching control and management. Such a switch is typically controlled by a host, the audio server, which manages the switch and may have an interface to the switch in the PSTN for out-of-band signaling.

PBX conferencing: In an enterprise environment, a natural place for audio conferencing is the PBX. The PBX technology has advanced from analog to digital audio mixing, rather than audio bridging. Mixing requires digital signal processing (DSP) hardware to process the audio streams. Whereas, bridging simply switches the streams without processing.

CTI PC-based conferencing: A small scale audio-conferencing system may be built using a PC with *computer telephony interface* (CTI) cards interfacing to the PSTN and conferencing cards having DSPs for voice processing.

In order to allow for multiple platforms for audio-conferencing, yet preserve a common, generic audio control interface to the MMDC *conference control*, we adopted the modular/layered architecture shown in figure 3. Depending on the conferencing platform, an appropriate manager/driver is selected. On top of such manager/driver, we develop a platform specific controller which brings the platform to a common switching application programming interface (API). Then, the ACM acts as the interface layer between MMDC and the audio domain.

We have opted to use TCP/IP connections for communicating commands between the MMDC conference control and ACM, as well as between ACM and audio controllers. Thus, if this is appropriate, we can place the ACM on a machine which is separate from either the MMDC server complex, or the switching hardware. This decision of ours has simplified the development and implementation of ACM. Thus, the ACM can be built independently of the "core" MMDC development platform, and leverage other applications that may require audio-conferencing control capabilities as well.

Each MMDC "start_a_conference" request initiates an ACM server process, hence multiple ACM server processes are activated when more than one conference is started. Each ACM server process handles one conference and manages the following important tasks: add/drop participants from the audio-conference; respond to MMDC queries, e.g., about status of telephone connection, the status of the switching application, etc; send queries to and receiving responses from the switching hardware platform; poll the switching hardware for conference status and send conference status messages to MMDC, etc.

4.2 The MMDC client

The MMDC client has been designed around three main components: (a) the web browser; (b) Java applets embedded in HTML scripts to be loaded on the web browser; and (c) a Java application with server functionality, which we call the *client-side server*. Figure 4 sketches some high level details of the client architecture. Once again, the MMDC client has been designed to be platform independent, hence, the entire client is Java based.

4.2.1 The web-browser

The web-browser is used by the user to connect to the MMDC server complex and request to join or create a conference. The server web-page and menu selection pages guide the

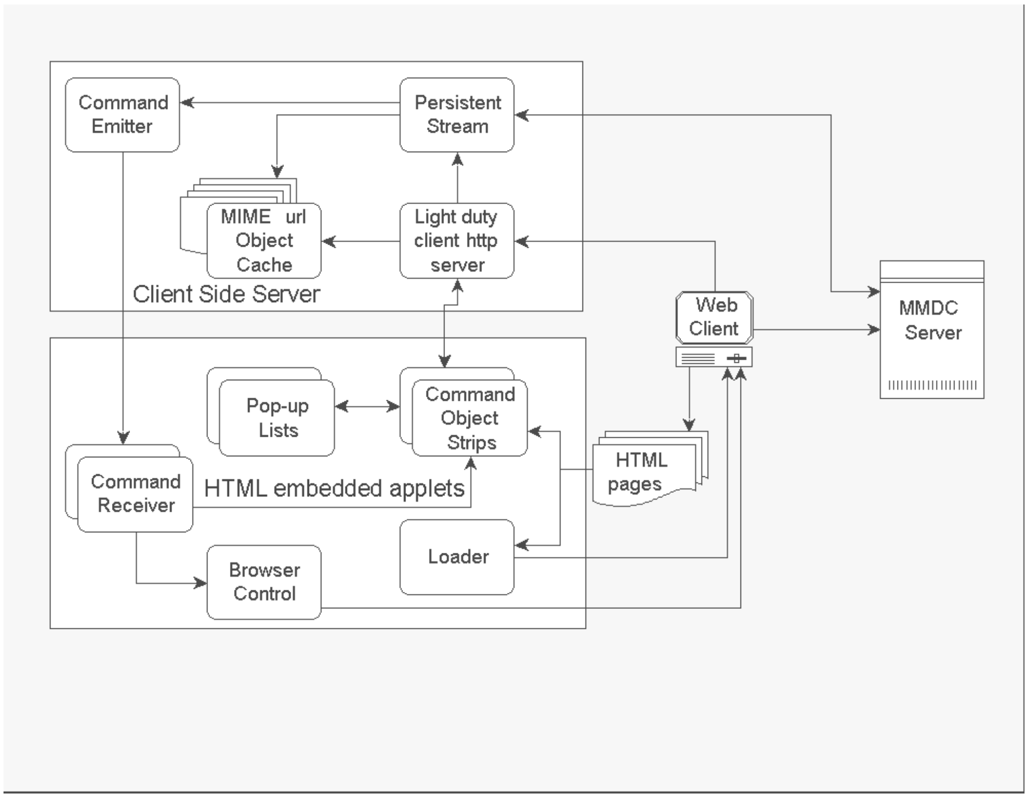

Figure 4: MMDC client architecture

user through this initial phase. When the MMDC *concierge* accepts the request of joining a conference, it causes the web-browser to load a page of frames embedded with a few applets that will define the client GUI interface and establish the persistent communication with the MMDC server.

Figure 5 is the interface for chairperson of a conference at this stage of execution of the client. The main window of the browser is divided in three horizontal frames, e.g., see figure 5. Currently, the top and the bottom frames offer access to a user to a set of commands that will be described in the rest of this section. The central frame is dedicated to the display of the HTML documents distributed by the chairperson of a conference to all the participants of the conference.

4.2.2 The Java applets

We have currently implemented two kind of control applets:

Control strip applet: This kind of applet allows to be built a dynamically changeable GUI. When loaded by the browser, it creates a set of buttons whose shapes, labeling and associated functionality are described in a configuration file. The MMDC server has the capability of sending commands to this applet that have the effect of changing any of the components and characteristics of the interface: button label, associated action, or even remove or add a new bottom. Eventually the full interface can be dynamically changed on-the-fly replacing the associated configuration file. Currently the MMDC server uses these dynamic capabilities of the control strip applet to indicate the number of participants currently in the conference. Any time a user joins or leaves the conference the label of the button used to display the list of participants changes to indicate the actual number of participants. Figure 5 shows the screen of the chairperson during the conference after he has requested the participant and slide-tray lists by pressing the corresponding control strip buttons.

Interface manager applet: This is a very simple applet that is used to force the browser to retrieve HTML pages from the HTTP interface of the local client-side server. The pages are displayed in the central frame of the client interface of figure 5.

4.2.3 The client-side server

This is the central part of the client architecture. It is used to perform the following essential functions.

1. Initializing the main GUI part of the client, see figure 5.
2. Provide communications means with the GUI applets.
3. Create a persistent connection with the MMDC server to use for exchanging commands and data that will have to be displayed to the user.

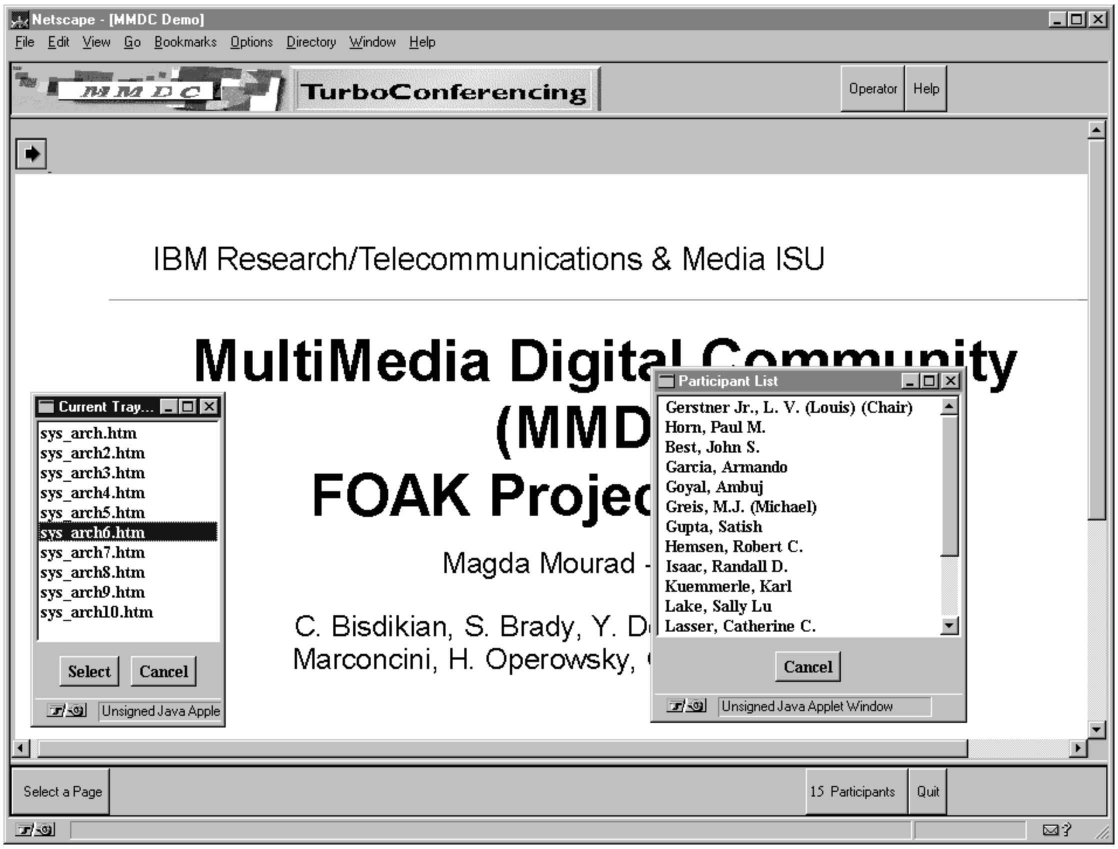

Figure 5: MMDC client GUI

4. Cache data coming from the MMDC server.
5. Offer a minimal subset of an HTTP demon interface used to make available the cache files to the web browser. These files include:
   - a set of Java classes and HTML files used to boot the client interface on the browser;
   - all the new HTML documents and attached graphics sent from the MMDC server to be displayed on the client;
   - the *user interface markup language* (UIML) scripts used to configure the GUI of control strip applets

Following the acceptance of a user by the *concierge*, the browser will load the user interface components from the HTTP demon interface of the client-side server and ask the client-side server to activate a persistent connection with the MMDC server specified by the *concierge*. At this point the client software is fully active.

The client-side server communicates with the interface manager applet to execute the DISPLAY command to display a cached HTML page on the client browser. It will also communicate with control strip applets to execute the MODIFY, REPLACE, ADD, DELETE commands to "customize" appropriately the command strip interface. A PUSH command will be generated by the client-side server of a chairperson any time that he selects a new slide from a slide tray, or he clicks on an active link of an HTML page previously displayed by the MMDC server. As a result, the selected slide or linked page will be displayed on each conference participant's web-browser. The client-side server will send a LEAVE command to the MMDC server whenever the user clicks on the QUIT button of the interface.

5 Discussion

5.1 Competing solutions

The principles and design objectives of MMDC, namely a simply to use intergrated data-/audio-conferencing application that is platform independent and scalable, set MMDC apart from many other solutions in the collaborative applications arena, that require either homogeneous communications, or operating system environments. In this section, we will present some of these other solutions, while an extensive collection of Internet-based collaborative applications, see [6].

In the recent past, there have been a number of trials conducted and offerings announced that overlap in various ways with MMDC. Many of these have been discussed on the web-sites of the parties involved. As expected, TelCos have been prominent in this arena. Both AT&T [1] and Sprint [7] have created web interfaces to their teleconferencing services that can be used to reserve or even control an audio teleconference. However, this control function is separate from 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, [2]) and in the case of Sprint, the data conferencing relies on Quarterdeck's TALKShow client software. In either case, the assumptions are that in order to participate, clients should have downloaded and installed the prerequisite client code, and while platform interoperability is assured if the client software is available for multiple platforms, interoperability across trials or even with different "standards compliant" clients is not possible.

MMDC covers a richer set of connectivity options, including a two-line solution, which guarantee 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 participant would also be able to participate in MMDC sessions.

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

There are also a variety of Internet collaboration tools that are becoming available that feature voice transport over the Internet [6]. Netscape's CoolTalk or Microsoft's NetMeeting are among the more visible examples. However, since the Internet as it currently exists cannot offer the quality of service guarantees needed to ensure high quality voice, MMDC's use of the PSTN for voice transmission is an advantage. Note that we allow direct control of telephone switches from within the MMDC workspace, for high-quality voice multi-party conferencing. No Internet collaboration tool provides such a functionality today.

Other solutions are being offered by software companies large and small. Some see this application as a way to build the market for other products they offer such as T.120 and H.320 toolkits (cf. MCI/DataBeam, AT&T WorldWorx800 Personal Conference Service). These solutions are based on the ITU standards to allow cross platform interoperability and support, e.g., Intel Proshare, Apple QuickTime Conferencing, DataBeam, etc. These solutions allow real time interactive data, voice, and also video conferencing. These solutions connect locations with dissimilar line speeds and video equipment to participate in a collaborative session. They can run over one (multiplexed) or more (separate) 800 number links with various speeds, e.g., from 300 bps to basic rate ISDN (56/64 Kbps) to prime rate ISDN (112/128 Kbps) to T1 (1.5 Mbps). The TelCos offering these solutions provide these links and support services (e.g., reliability, customer assistance, international access, etc.) This means that special hardware at the users' 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 disadvantage is the higher price, as well as the special hardware and client software (platform dependent) needed to allow multiple users to collaborate.

5.2 Other MMDC applications

While the initial focus application of MMDC is restricted to 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 the most obvious additional applications of MMDC are in distributed education, where the existing MMDC system may already provide a good fit with only minor modifications. Using the existing MMDC system, an instructor can already convene a class, display lecture material using web pages, and interact with the class concerning that material on the audio channel. Students should be able to indicate that they have questions using the existing MMDC user interface, and the instructor should be able to 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 plus 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 is in a range of meeting formats that are currently almost entirely done face-to-face. Town meetings, including school board meetings, judicial proceedings, committee meetings, and many others, need not be restricted to those with the mobility to travel to those meetings. By combining audio conference access to meetings with data channel ability to view meeting presentation materials, remote participants can be brought into existing meetings. It is a small step, moreover, to conduct these meetings on an entirely distributed basis. This kind of environment may be enhanced by giving participants a choice of several views of the meeting environment, each of which is distributed as discrete data "tracks" in the MMDC environment.

Yet another application, the *multimedia group shopping*, is actually the original application from which the idea of MMDC grew, and is perhaps its biggest 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. This is an important variant of the shopping experience that has so far been ignored by web and other on-line service providers. In today's shopping world, large numbers of people shop in groups, using shopping as an excuse to get together. These groups of social shoppers have attracted relatively little attention among those who are developing electronic commerce for on-line services and the Internet. Interesting variants of multimedia group shopping include: (a) scheduled shopping trips, perhaps involving friends, relatives, or others; (b) public shop and chat sessions, in which an initiator can declare a topic and or shopping focus, and anyone can join in; (c) celebrity shop and chat sessions, in which an advertised celebrity takes people on a shopping tour on a scheduled (perhaps periodic) basis; (d) electronic auctions, in which the main browser window displays the item at auction, the MMDC server "auction application" accepts written electronic bids across the network, and the interaction is dominated by the auctioneer's patter and discussion of new items, etc.

Even more applications can be envisioned with additions such as including agent monitoring and moderation of interaction, 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. Audio conference control, 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 in audio teleconferencing. 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 that combine all three. Working only with the data channel, the MMDC server, persistent stream services, and client interface can enable advertiser supported web services in which most of the browser window can view whatever the user points to, but a single control strip at the top or bottom of the window shows advertising and allows the user to modify their advertising profile. This kind of service might allow inexpensive or free access to the Internet in return for viewing an advertising stream. Finally, the user interface itself can be readily reconfigured to standalone intelligent network applications, including smart news clients that do cache ahead of the posts that a user is likely to ask for next and smart web search agents that search the web based on user criteria and add "hits" to the MMDC client server cache.

6 Summary

We have presented the MMDC project and prototype which is a client/server web-enhanced audio- and data-conferencing solution. Our approach to this highly active research and business arena was to develop a system that will be easy to operate, scalable, and universally available, i.e., platform independent. To resolve these problems, the MMDC client code is written in Java in order to allow the same code to run on different operating systems. Also, a conference participant is not required to have special hardware except a modem or LAN adapter. The client software uses the Internet protocols to communicate with the MMDC server to allow interoperability between conference's participants as well as Internet access, and international access.

In MMDC, we allow the users to use the Internet to access phone directories, make conferencing reservations, change these reservations, etc. We allow conference chairpersons to push web-page content to clients very efficiently, in the form of preloaded HTML slides, or just retrieved information from the web, in a way that guarantees a consistent web-page view across all the conference participants. In MMDC, we have integrated audio control facilities that allow for the establishment, and control, of high-quality voice-conferencing among the conference participants.

The MMDC project, funded by IBM's FOAK program, has been developed with the audio-/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, electronic auction events, etc.

Acknowledgements

The authors of this paper are deeply indebted to their colleagues Collin G. Harrison, Jurij R. Paraszczak, Eddith H. Stern and Barry E. Willner for their ideas, leadership and support they so generously provided to the MMDC project.

References

[1] AT&T TeleConference Service, at URL: "*http://www.teleconference.att.net.*"

[2] B. Aldred, *Desktop Conferencing: A complete guide to its applications and technology*, McGraw Hill, 1996.

[3] *Forward Concepts*, "Teleconferencing Markets and Strategies: the Convergence of Price, Performance, and Plumbing," at URL "*http://www.fwdconcepts.com/press01.htm.*"

[4] *IETF Hypertext Transfer Protocol (HTTP) Working Group* at URL "*http://www.ics.uci.edu/pub/ietf/http/.*"

[5] *Internet Telephony Resource List*, at URL "*http://rpcp.mit.edu/ itel/resource.html.*"

[6] *PC Magazine*, special report on "Internet Collaboration," Oct. 1996.

[7] Sprint Internet Conference Center (SM), at "*http://www.sprintconf.com.*"

[8] W. Stallings, "The backbone of the web," *Byte*, pp. 63-64, Oct. 1996.

[9] *The Common Gateway Interface* at URL "*http://hoohoo.ncsa.uiuc.edu/cgi/overview.html.*"