IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

Windy City Innovations, LLC, a Delaware Company,	)	
Plaintiff,	)	**Case No. 04 C 4240**
v.	)	**Hon. Samuel Der-Yeghiayan**
America Online, Inc., a Delaware Corporation,	)	
Defendant.	)	

EXPERT REPORT OF BRUCE M. MAGGS

1. My name is Bruce Maggs. I have been retained by the defendant in this action, America Online Inc. ("AOL") to consult on technical issues pertaining to this lawsuit and to prepare a report that provides a summary of the testimony that I am prepared to give at trial, if called to testify. This document constitutes my expert report on the validity and enforceability of U.S. Patent 5,956,491.

2. In summary, frist, this report explains my opinion as to why the '491 patent is invalidated by the prior art. It also explains my opinions that the patent fails to disclose the claimed invention's "best mode." Finally, the report indicates my opinion that the Gtalk software, co-authored by the named inventor prior to the invention, but not disclosed to the patent office, is non-cumulative.

3. My *curriculum vitae* is attached hereto as Exhibit A. In summary, in academia, I am a tenured Professor of Computer Science in the School of Computer Science at Carnegie Mellon University. I joined the faculty as an Assistant Professor in January 1994, was promoted to Associate Professor in July 1997, was given tenure in July 1999, and was promoted to (full)

Professor in 2004. I also held the position of Visiting Associate Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology from September 1998 through January 1999.

4. With respect to my industry experience, I helped launch Akamai Technologies in 1998. Akamai provides content delivery services for many of the world's most popular websites. I served as a Senior Research Scientist for Akamai from January 1999 through March 1999, and as Vice President for Research and Development from April 1999 through December 1999. I am currently the Vice President for Research at Akamai and have held this position since January 2000. In addition, I was a Research Scientist at NEC Research Institute, Inc., from September 1990 through January 1994.

5. I received my Doctorate degree in Computer Science from the Massachusetts Institute of Technology in 1989, my Masters of Science degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1986, and my Bachelor's of Science degree in Computer Science from the Massachusetts Institute of Technology in 1985.

6. Additional information concerning my teaching experience, publications, surveys, manuscripts, distinguished-lecture-series speeches, keynote addresses, invited lectures, awards, grants/contracts/fellowships, committee service, technical advisory boards, and the patents for which I am a named inventor, is set forth in my Curriculum Vitae.

7. I was also employed as a computer programmer at the University of Illinois at various times between 1979 and 1983. At the University, I wrote numerous programs for the PLATO computer system, including educational programs and recreational programs. One of these recreational programs was a multi-player "dungeons and dragons" game (or "MUD") called Avatar. Avatar, among its other features, included communications functionality that

allowed for messaging to a number of users simultaneously as well as messaging between two individuals. These messages could include both text and graphical images. PLATO is now known as NovaNET. I have played Avatar running on NovaNET over a public TCP/IP network.

I am familiar with many computer communications programs, including numerous "chat" and messaging systems. I am familiar with e-mail standards and protocols such as SMTP, POP, IMAP, and MIME, and have taught courses at Carnegie Mellon University on these standards and protocols. I am familiar with and have used other PLATO programs preceding Avatar, including "empire," "talkomatic," and "term talk," which provided similar communications functionality.

8. I have testified before as an expert witness in the lawsuit captioned *Lexmark International, Inc. v. Static Control Components, Inc.*, No. 02-571-KSF, United States District Court for the Eastern District of Kentucky.

9. I am being compensated at the rate of $300 per hour for my work in this case.

10. In preparing this report, I have thoroughly reviewed a number of documents and other materials, and have otherwise prepared for the report as discussed below. The pertinent documents have been attached as exhibits or are included in the attached CD-R and DVD-R. The documents that I have reviewed include:

- U.S. Patent 5,956,491, and the documents that comprise the "file history" of this patent (including the references cited therein) (Ex. 1).
- The source code appendix to the '491 patent. (Ex. 2).
- An America Online service called "Road Trips." I reviewed versions 1.3, 1.30, 1.64, 2.0, and 2.1 of the primary source code file for Road Trips, which was called "tour.c." (Ex. 3), I also reviewed CVS logs for the files tour.c (Ex. 4) and tour2.c (Ex. 5), and a set of printed "screenshots" of the forms used by Road Trips. (Ex. 6). I also reviewed AOL 2.5 client software (Ex. 7) and a list of forms and form creation dates. (Ex. 7).

- Certain Netscape Communications software known as "Netscape Chat" version 1.0.1.8 (32-bit) and 1.01 (16-bit), and Netscape Navigator version 1.22 (16-bit), (Ex. 8) and associated source code and design specifications. (Ex. 9). I also compiled and configured an IRC server, called ircd, from Undernet, the source code for which is attached. (Ex. 10).
- Network Working Group Request for Comments (RFC) 1459, "Internet Chat Relay Protocol", by J. Oikarinen and D. Reed, May 1993. (Ex. 11).
- Sun Microsystems' HotJava Browser, the applet viewer from Sun's JDK version 1.0, and Netscape Navigator version 2.0 (Ex. 12).
- Certain CompuServe software entitled CompuServe Producer, as well as CompuServe's "WINCIM.EXE" client program, and various associated source code files. (Ex. 13).
- Several versions of software known as Gtalk, including versions 1.6.8, 1.6.6, and 1.6.4 for the Unix operating system; version "1.9z1.4" for DOS, and version 2.2.3 for OS/2, which I understand was prepared in part by Daniel Marks, the named inventor of the '491 patent, or were derived from software written by Mr. Marks, and "GTUX," another software program. (Ex. 14).
- The "Gtalk Owners Manual" dated July 14, 1995. (Ex. 15).
- An article by Prof. Judith Donath and Niel Robertson, entitled "The Sociable Web," posted on the World-Wide Web in October of 1994 (Ex. 16) and the following related documentation (Ex. 16):

DX017	Sociable Web Article
DX018	Sociable Web Article (no pictures)
DX019	World Wide Web Conference pamphlet
DX021	Sociable Web Article HTML source code
DX024	World Wide Web Conference pamphlet
DX086	README.TXT file

- Online versions of "The Sociable Web," found at http://smg.media.mit.edu/people/Judith/SocialWeb/SociableWeb.html and at http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/CSCW/donath/SociableWeb.html (Ex. 17),, and the HTML source files for each page (Ex. 18), and also online files found in the web directory http://smg.media.mit.edu/people/judith/SocialWeb/Pix/ (Ex. 18).
- A transcript of the May 25, 2005, deposition of Niel Robertson. (Ex. 19).
- The Mosaic User Authentication Tutorial, http://hoohoo.ncsa.uiuc.edu/docs/tutorials/user.html, dated 9-27-95. (Ex. 20).

- Upgrading NCSA HTTPd, http://hoohoo.ncsa.uiuc.edu/docs/Upgrade.html, dated 08-01-95 (Ex. 21).
- NCSA Mosaic Version History, http://www.ncsa.uiuc.edu/Divisions/PublicAffairs/MosaicHistory/history.html (Ex. 22).
- A log entry from November 18, 1993, on the NCSA Mosaic Website, indicating that Mosaic 2.0 was available on that date, http://archive.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/old-whats-new/whats-new-1193.html (Ex. 23).
- An article by Markus Sohlenkamp and Greg Chewlos entitled "Integrating Communications, Cooperation, and Awareness: The DIVA Virtual Office Environment." Proceedings of the Conference on Computer Cooperative Work, October 22-26, 1994. (Ex. 24).
- U.S. Patent 5,880,731 (a Microsoft patent) (Ex. 25).
- To the extent not encompassed in the above, the contents of the production CDs that bear production numbers DM 50-53, and WCI 001589, 002859, and 002860, which include additional source code not included in the patent. These documents have not been attached to this report per the protective order.

11. I have otherwise prepared for this report as follows:

12. I reviewed and analyzed the C programming language source code contained in the '491 appendix. I obtained an electronic copy of a transcription of this code and prepared executable software based on this code. When compiled, the source code produces two executable files, a "server" program called "uc" and a "client" program called "ucc". Both of these programs are meant to execute on the same computer. (The client program connects to a server program running on a machine named "localhost", which refers to the same machine.) I also connected to this computer running the executable software over a public TCP/IP network connection using the telnet application running on a different computer, and observed the behavior of this compiled software. I studied the client and server executables running on a computer with the Red Hat Linux 6.2 operating system, and also on a computer with the Linspire Linux operating system.

13. I also performed tests in which connections were made to the server program compiled from the C source code appended to the patent using a client program implemented as a Java applet found on the CD bearing the production number DM52, and executed on a different computer. The client program connected to the server program using a TCP/IP connection.

a. All tests with the Java applet were performed on an isolated private network located in the offices of Banner & Witcoff consisting of a desktop computer running the Linspire™ Five-0™ distribution of the Linux operating system and a laptop computer running the Windows XP operating system. The two computers were both connected to a Linksys router, model number BEFSR411 ver.3.1

b. Java code is run using what is known as a "Java virtual machine" (JVM). The Java code on the laptop was run using a number of different JVMs. First, I used the Java applet viewer provided in Sun Microsystems Java Development Kit (JDK), version 1.0. Next, I installed the Apache web server on the Linspire server, and prepared an html document that instructed a web browser to fetch and then execute the Java applet from the same web server. The Java applet was then tested by "downloading" both the html document and subsequently the applet using both Sun's HotJava browser, and also Netscape Navigator, Version 2.0.

c. When using the Java client, it is not necessary for the client executable generated from the C code appended to the patent to also execute on the server machine. However, to test the interoperability of the Java client

and the C client, I connected to the server simultaneously using a telnet session and using the Java client.

14. I reviewed AOL Road Trips.

a. I tested America Online's "Road Trips" service, version 2.0, and discussed the operation of this service with one of the creators of Road Trips, Jay Elinsky. To experiment with Road Trips, I registered an account with America Online (AOL), brucemmaggs@aol.com, and then connected via TCP/IP to an AOL server using version 2.5 of the AOL client program, running on a Pentium 75 MHz desktop with the Microsoft Windows 3.11 for Workgroups operating system.

b. The installation of the AOL client software version 2.5 was created using a copy of the contents of an AOL CD-ROM containing installation software provided by AOL. I did not have a copy of an original AOL CD-ROM. The CD-ROM version, which has a much higher storage capacity than a floppy disk, installed a copy of IWENG.DLL with a creation date 8/30/1995. Its length was 700KB.

c. I also experimented with different versions of version 2.5 of the AOL client program on various computers, and saw no differences in behavior.

d. The first version was installed on a desktop computer (the participator computer) using an original AOL floppy disk titled "America Online FOR WINDOWS™ Version 2.5". America Online produced these floppy disks in 1995. The disk contains a single file called "SETUP.EXE", which, when executed, installs the client software, which consists of several files.

The creation dates of the files copied from the disk are all 6/27/1995 or earlier and the copyright notice on the disk label reads "©1993-1995 America Online, Inc.". The installation program SETUP.EXE also creates several new files and directories (e.g., main.idx) whose creation dates are set to the date of the installation, e.g., 6/23/2005. None of these files are executable code or modify the operation of the software. The computer was running the Windows XP operating system.

e. The floppy disk did not install two files, "IWENG.DLL" and "TOOL/WWW.AOL" that are required in order for the browser/chat window of Road Trips to operate correctly on the client computer. These files were not included in SETUP.EXE on the floppy disk because IWENG.DLL is large (698 kilobytes) and would not fit on the disk with the other files, even when compressed. (The capacity of a floppy disk is only 1.44 megabytes.) AOL client version 2.5, when installed from this floppy disk in 1995, would instead automatically retrieve the files from AOL when they first used the software to access AOL using a dial-up connection or using TCP/IP. Today, however, while it is possible to connect to AOL using version 2.5 of the client software, version 2.5 is no longer fully supported, and I could no longer download these files. For this reason, Jay Elinsky provided me with copies of these files with a creation dates of 6/27/1995. He indicated that the files were taken from an old laptop computer on which AOL client software version 2.5 had been installed around 1995. I then installed these files in the same directory as

the other AOL software and then accessed AOL by starting the program AOL.EXE.

f. I also tested the AOL client software version 2.5 installation that was copied in its entirety from Mr. Elinsky's old laptop computer. The latest creation date on any of these files was 6/27/1995.

g. In some cases, I made one change in a configuration file called "TCP.CCL". This file specifies the host name and port number to connect to when accessing AOL using TCP/IP over the Internet, which was possible using version 2.5 of the client software. The host name in the original file is "AmericaOnline.aol.com". I was able to access AOL and use Road Trips using this hostname. The Road Trip service is installed on a server that is accessible via TCP/IP connectivity to a "BERP" server, which sometimes is not assigned when connecting to americaonline.aol.com. To ensure connectivity to a BERP server, I modified the line

```
NetConnect  1  5190  10  AmericaOnline.aol.com
```

by changing it to

```
NetConnect  1  5190  10  berp-nz01.dial.aol.com
```

Jay Elinsky informed me that he had installed version 2.0 of the Road Trips software on today's production AOL system. The software is run on an AOL server. All of the code that is used to implement Road Trips is identical to the code version 2.0 as written in 1995. Mr. Elinsky added one line of source code to assist the software in operating in the current AOL environment. Specifically, the following single line was added to

tour.c in function "Initialize_Application":

```
AFI_Set_Per_User_Stream_IDs  (/*SID_CHAT_LO*/ 0x200ac,
/*SID_CHAT_HI*/ 0x0400ab);
```

This line does not change the functionality of the Road Trips code. Instead, this line assists the Road Trips code in operating in the current AOL environment.

h. In addition to source code, AOL services make use of "forms". A form is a graphical script that controls the graphical user interface provided by the AOL client software. A form is interpreted by the AOL client software by the participator computer. It specifies, for example, where a button should appear on the screen and what should be transmitted to an AOL server when the user clicks on the button. A form also specifies where a browser window should appear, where a field for entering text should appear, and where a scrolling text field should appear. An AOL server can send an "atom" to an element of a form in order to change its appearance or behavior. For example, an AOL server can send an atom to a button created by a form indicating that it should no longer appear on the form.

i. Road Trips used various forms. Some of these forms are depicted in printed screenshots in Ex. 6. Jay Elinsky informed me that he installed these forms on today's AOL production system so that they could be sent to the AOL client software when users today access Road Trips. Elinsky also indicated that all of the Road Trips forms used today are dated before August of 1995. A list of forms and form creation dates which confirms this was given to me by Elinsky and is attached as Ex. 7.

j. After accessing AOL via a TCP/IP connection over the public global TCP/IP series of networks, using the version 2.5 of the client software, I then pressed control-k, which caused an AOL "form" to appear on my screen that allowed me to enter an AOL keyword. I was provided with an AOL keyword that led to Road Trips. Jay Elinsky also connected to Road Trips at the same time.

k. Jay Elinsky and I then participated in several tours, and exercised the features of the software. One difference between my account and Mr. Elinsky's account is that mine is a "member" account, whereas his is an "internal" account. Hence his account has certain privileges that mine does not. Mr. Elinsky was able to create an "AOL tour", a tour that a user with a member account cannot create, but that users with "internal" or "overhead" accounts can create. I accessed this "AOL tour" and observed the behavior of Road Trips during this tour. I created a "member" tour and participated in this tour with Mr. Elinsky, and observed the behavior of Road Trips during this tour. Another type Road Trips tour is a "private tour." Mr. Elinsky demonstrated the creation of a "private tour." I joined a private tour created by Mr. Elinsky and observed the behavior of Road Trips.

l. I later created a video record which demonstrates the features of Road Trips while using computers built with parts available in 1994 (in particular, systems based on the Intel P75 processor) and running the

Windows 3.11 for Workgroups operating system. A copy of this video record (on DVD) is enclosed as Ex. C.

15. I reviewed Netscape Chat.

a. I used "Netscape Chat" version 1.01 (16-bit) on the same Pentium 75 MHz computers running Microsoft Windows 3.11 for Workgroups operating system. Netscape Chat was installed using an original Netscape "Power Pack" CD-ROM. This CD-ROM contains a program called Netscape Power Pack™ (Powerpack.exe) that allows a user to install certain software, including Netscape Smartmarks™, Netscape Chat™ (version 1.01), Adobe™ Acrobat™ Reader (version 2.1), Apple® QuickTime® version 2.0, and RealAudio™ Player (version 1.0.0). Power Pack can be used to install either the 32-bit version of Netscape Chat, version 1.0.1.8 (for use with Microsoft Windows 95 and later versions of Windows), or the 16-bit version, 1.01 (for use with Microsoft Windows 3.1). Netscape Navigator version 1.22 (16-bit version) was also installed on the same Pentium 75 MHz computers. I examined the behavior of the Netscape Chat program (and simultaneously the Netscape Navigator program) by making connections from both machines using a TCP/IP over the public Internet to an IRC server that I set up.

b. I later created a video record in which I demonstrate the features of Netscape Chat while using the same computers and operating system. To perform these experiments, I downloaded the Undernet IRC chat server software ircu2.9.19 from ftp.undernet.org. A copy of this software is

attached on a CD-R as Ex. 10. The IRC chat server ircd was then configured, compiled and run on a Sun Microsystems SparcStation 4 workstation, using the Solaris 1.1.2 operating system, which was installed from an original Sun Microsystems CD-ROM. Both the workstation and the operating system were available in 1994. To compile the IRC server, I made two syntactical changes to configure the code to compile on the Solaris C compiler; in 1994 and 1995, different C programming language compilers supported slightly different syntaxes, and this change would have been normal to a programmer of that time in order to allow this software to compile on Solaris.

i. added a comma at the beginning of line 1704 in file ircd/s_bsd.c.

ii. added a comma at the beginning of line 726 in file ircd/s_user.c

c. To configure the IRC server with a resolvable host name, and to verify that the connections between the IRC server and the IRC client computers were made through the Internet, I moved it to a location remote from the Pentium computers and connected the server to a different Internet Service Provider (ISP). The IRC server was placed behind a firewall whose public name was irc.mooreusa.net, and whose public IP address was 64.81.139.232.

d. I also configured the IRC server by creating a file ircd.conf from example.conf, a file sample configuration provided with the IRC server software. I modified example.conf to change the name, port number, and

access password for the server, and to remove a number of non-mandatory configuration lines.

e. I also reviewed the C++ source code for Netscape Chat and associated design documents and specifications, as well as the Internet Relay Chat (IRC) Specification, Request for Comments (RFC) 1459.

16. I reviewed the CompuServe Producer / Viewer system and CompuServe's CB Conferencing system.

a. I tested "CompuServe Producer" V.198C, Copyright 1996, using the following hardware configuration. The Producer software, program csprod.exe, was run on a Packard Bell personal computer. A sticker on the computer indicates that its model is "LEGEND 2150 50MHz DX/50 486 processor PC with Microsoft Windows operating system." The operating system installed on the machine was Microsoft Windows 3.1. A fact sheet that accompanied the computer also listed the model as "LEGEND 2150 MULTI-MEDIA". The fact sheet gave a "Test Date" of 10/20/93 01:26:27PM. The fact sheet indicated that the sound card was an SGPRO-16, but I noticed that the Windows software called the sound card an MM 16 PRO. A video capture card that did not come with the computer had been installed in the computer. The original box containing the video card indicated that it was "PCVD1000 Intel Smart Video Recorder for Indeo™ Video". There was a shipping date on the box that read "6/10/94". The specifications of the card are listed on the back of the box. They indicate that the card is a full ISA board, that it has one RCA

and one S-VHS video input jack, and that it accepts NTSC or PAL analog video-composite Y/C (S-VHS). Finally, the video capture card was connected to a Mitsubishi VCR HS-U590 video cassette recorder machine. In particular, the VIDEO OUT (yellow) jack from the VCR was connected to the S-VHS video input jack on PCDVD1000 using an RCA video cable. The audio output jacks on the VCR were not used. Instead, a microphone was connected to the 3.55mm miniature input jack labeled "MIC" on the sound card. The serial port on the computer was connected to an external dial-up modem.

b. I executed a program called csprod.exe. This program immediately brought up a window labeled "CS Producer". I then selected "GO" from the "Session" drop-down menu. I then entered "CATHOLIC" for service. This is a current service (a place for users with shared interests to gather and chat) on CompuServe. I had the option of pressing "Set Nickname" to choose a nickname other than my CompuServe user ID (which is just a number), so I chose "Maggs." I then clicked on "Go". At this point the Producer software dialed in to CompuServe using a modem, making a connection was made to the CompuServe server that hosts the "CATHOLIC" service.

c. Once I was connected, a "Room Selection" window came up. I selected "9 – Music Room", and a "CompuServe Control – Room 9" window popped up. This window had two sections, one labeled "Image Control" and the other labeled "Audio Control."

d. Included in the Image Control section was a button for "Send Image Snapshot." There were also check boxes for setting the video quality to low, medium, high, and very high. There were more check boxes for "Auto Send Images" and grayed out (not active) "Send Closed Caption." Under "Audio Control", there were buttons for "Record", "Play", and "Send". There were also check boxes for setting the audio quality to low, medium, good, and radio.

e. The "CompuServe Control - Room 9" window also provided buttons for "Chat…", "Users…", "Change Room…", and "Select Handle…" Pressing Chat pops up a "CS Producer – Chat Window" box, with a scrolling dialog box and text entry field. Pressing Users pops up a "User List" window, which shows the other participants in the same room. Pressing Change Room pops up a window with a list of other rooms in the same service. Finally, pressing Select Handle pops up a window that allows a user to his or her nickname (apparently the terms "handle" and "nickname" are used interchangeably.

f. At the same time that the "CompuServe Control – Room 9" window came up, the title of the "CS Producer" window changed to "CS Producer (on line)", and the video playing from the VCR appeared in this window. The window provided three pulldown menus, "Session", "Options", and "Help". Under Options the choices were "Video Format…" and "Video Source…" I selected "Video Format…", and this opened a "Video Format" window. Here there were pulldown menus for "Video

Compression Method:" which I set to "Intel Indeo[TM] R3.1 Video", and "Size:" which I set to 160x120, indicating 160x120 pixels. I then selected "Video Source..." and a window popped up. In this window, for "Input Source" there were check boxes for "Composite" and "S-Video (Y/C). I selected composite because that is the format provided on the VCR's VIDEO OUT jack. Under "Input Type" there were check boxes for "NTSC" vs. "PAL." I selected NTSC, as that is the type of the signal on the VCR's VIDEO OUT jack. Finally, I clicked a separate button labeled VCR.

g. At this point, on another computer, a desktop running the Windows 2000 Server operating system, I started the "CompuServe Information Manager" (CIM) software, by executing a file called WINCIM.EXE. CIM is the standard client software run by CompuServe users. WINCIM.EXE includes executable code called Viewer that implements the client side of the Producer / Viewer system on the participator computers that are not running the Producer software. I clicked on the green traffic light "go" button, and was asked for a room. I was asked to select a service, entered "CATHOLIC", and was logged into CompuServe via TCP/IP with a different user name.

h. Once logged in, I began by pressing the "Who's Here" button. A "Who's Here" window popped up, and I was able to list all users in the service or all users in any particular room. I observed that Maggs was in Room 9. Next I pressed the "Enter Room" button, and selected Music (9). The

rooms have both names and numbers, and room 9 is also known as the Music room. A "Music Room" window came up, which contained a scrolling dialog chat window with a line for entering messages. When this Music Room window appeared, I saw my other user name (which I had arbitrarily chosen to be "Phoebe") appear in the list for Room 9 in the "Who's Here" window.

i. On the producer side, as user "Maggs," I pressed "Users..." and a window labeled "User List (2)" popped up, showing two users in room 009. ("Maggs", running the Producer software, and "Phoebe", running client software, CIM). As user "Maggs," I pressed "Change Room...," which brought up same "Room Selection" window seen before. It listed nine different rooms associated with "CATHOLIC" service. User "Maggs", however, did not change rooms. As the user "Maggs," I then pressed "Select Handle..." and chose "Bruce" as a new handle. I noticed that the name changed on the user list. From this point forward, any chat messages sent by the producer were labeled "Bruce>" rather than "Maggs>."

j. As user "Bruce," I then pressed "Send Image Snapshot", and a "Bruce Image" window immediately appeared on "Phoebe's" screen, showing snapshot of video that was being played by the VCR. This message was sent by the Producer software to a CompuServer server running the CB Conferencing system and from there to the Viewer software on user "Phoebes"'s computer. Phoebe" then received a text message from

"Bruce" in her chat window. As user "Phoebe," I then sent a text message to the chat room, and "Bruce" received it. As user "Phoebe," I then pressed the "Ignore" button, and an "Ignore..." window came up. I selected "Bruce" from list of Room 9 users. As user "Bruce," I then entered a text message. This time it did not appear in "Phoebe's "chat window. As user "Phoebe," I then sent a message to the chat room. It did appear in "Bruce's" chat box. User "Phoebe" was ignoring "Bruce", but "Bruce" was not ignoring "Phoebe".

k. As user "Bruce," I then pressed "Record" in the "CompuServe Control – Room 9" window, and recorded a brief message. I then pressed "Play" and heard the recording, then pressed Pressed "Send" but did not hear it on "Phoebe's" computer because it did not have a sound card.

17. I compiled and ran Gtalk versions 1.6.4, 1.6.6, and 1.6.8 for Unix. The Gtalk source code produces two executable programs, a server program called "gtalk" and a client program called "gtclient". I ran and studied the software with both the server and client software running on a computer with the same Red Hat Linux 6.2 operating system. I connected to this computer over a public TCP/IP network using the telnet application, and I studied the behavior of each of these versions of GTALK.

18. I reviewed The Sociable Web.

a. I examined the two html versions of "The Sociable Web" paper by Donath and Robertson found at http://smg.media.mit.edu/people/Judith/SocialWeb/SociableWeb.html and at

http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/CSCW/donath/SociableWeb.html (Ex. 17) and the HTML source files for each page (Ex. 18) and also online files found in the web directory http://smg.media.mit.edu/people/judith/SocialWeb/Pix/ (printout at Ex. 18).

b. When I first examined the first document (Ex 17), hosted on the server smg.media.mit.edu, the embedded graphical images in the document (gif files), such as the one specified by the link below (found in the HTML source file (Ex. 18) <img src = "http://judith.www.media.mit.edu/SocialWeb/Pix/WhoOnlineText.gif">, the images did not appear in my browser because the gif files were not hosted on the server judith.www.media.mit.edu. The images, however, were available in the directory http://smg.media.mit.edu/people/judith/SocialWeb/Pix/ (printout at Ex. 18). In order to view the document with the embedded images, I prepared a local copy of the html document in which I modified the links to the images so that each referenced the host smg.media.mit.edu rather than judith.www.media.mit.edu. A view of the local html document, which shows the embedded images, is shown in Ex. 16.

c. Since my first viewing of the html document http://smg.media.mit.edu/people/Judith/SocialWeb/SociableWeb.html, however, as of this writing, the images have now been made available at judith.www.media.mit.edu, so that when viewing the document in a

browser, the gif images do appear. A view of this document, showing the images, in shown in Ex. 17.

d. In the second html document, the images do not appear, as the links to these images, such as the one below

<img src = "http://big-sleep.media.mit.edu:8000/SocialWeb/Pix/WhoOnlineText.gif">

refer to a server big-sleep.media.mit.edu that no longer operates a web server at port 8000.

19. I attended the deposition of Daniel Marks, the inventor named in the patent, on February 17 and 18, 2005.

20. I have also reviewed Windy City's positions with respect to claim construction as of January 5, 2005, contained in a letter dated January 5, 2005 (copy attached as Exhibit B). I also reviewed the parties' claim construction brief, dated June 30, 2005, entitled "Joint Brief on Claim Construction" (Ex. 27) as well as the Court's claim construction order of July 29, 2005 (Ex. 28).

21. All of the opinions in this Report are based on my personal observations and experience in this field. If called to testify at trial, I could testify based on observations and experience to all of the opinions presented herein. If called to testify at trial, I will be prepared to demonstrate all of the software that I tested, including without limitation compiled or otherwise operating code from the '491 patent and the code produced to AOL in this lawsuit, and to exhibit some or all of the source code.

22. In this report, some of my opinions pertain to obviousness. In evaluating obviousness, I understand that I must consider the following: (a) the scope and content of the

prior art; (b) the level of skill of a person of ordinary skill in the field at the time of the alleged invention; (c) the differences between the prior art and the claims; and (d) collateral factors such as failure of others to solve a technical problem, long felt need, commercial success of the process and other similar factors. Where I have provided an opinion of obviousness, I have used these factors, and I have considered the claimed subject matter as a whole in evaluating obviousness.

23. I believe that a person of ordinary skill in the pertinent art would have at least a bachelor's degree in computer science or a similar field (such as electrical engineering with a focus in computer science), coupled with at least three years of programming experience. I base this evaluation on my experience in this field.

24. At the time of the alleged invention of the '491 patent in April of 1996 (or in 1995, as alleged by Marks, the scope of the prior art would include prior art related to computer messaging technology. The content of the prior art includes many publications, patents, software products and services, conference presentations, and similar materials. I will discuss several specific prior art references in this report.

25. On the collateral factors, I note that there was no failure of others to solve the problem of the Marks patent, nor any long-felt need. Many others had created chat systems similar or identical to those of Marks, before Marks. Having attended the Marks deposition and heard Marks's testimony, I am aware that the Marks technology claimed in the '491 patent did not enjoy any commercial success. Few people ever used the Marks technology.

26. I have construed the claims of the patent in light of their ordinary meaning and in light of the Court's claim construction order of July 29, 2005. The parties differ on construction

of certain claims, and in such cases I have noted how one party or the other has construed a patent claim term.

INVALIDITY OF THE '491 PATENT OVER THE PRIOR ART

27. In this section nine pieces of prior art are examined. These pieces are

- AOL Road Trips
- Netscape Chat
- CompuServe Producer
- Gtalk
- The Sociable Web
- WebTalk
- DIVA
- U.S. Patent 5,880,731

28. America Online's "Road Trips" software, system, and service, contains all of the elements of the asserted claims in the '491 patent.

29. AOL Road Trips allows a user connected to AOL's service to lead other users on a "tour." A tour is the same as a "group" in the context of the '491 patent.

30. Upon accessing the Road Trips application, any AOL member could create a "member" tour or a "private" tour. AOL users with "internal" or "overhead" accounts could also create "AOL" tours. To create a tour, a user pressed a button that said "Create Tour." The user would then enter a title for the tour (twenty characters or less), and a slightly longer "description" of the tour. The user also had the option of entering URLs that might be shown to users during the tour. The user could then activate the tour by pressing a button for a member

tour, a private tour, or, if allowed, an AOL tour, at which point the creator became the tour "guide." The window for creating the tour would remain open during the tour, allowing the guide to store additional URLs, or to end the tour.

31. An AOL user upon accessing the Road Trips application could list the titles of the current AOL and member tours by clicking on a button that said "Active Tours." The user could then join a listed tour, or see a longer "description" of the tour that had been entered by the creator, or see the members of a tour. Private tours were not listed. To join a private tour, the user would have to know the secret name of the tour, and could join the tour by typing that name. Hence the name of a private tour served as a "password" to join the tour.

32. Once activated, users participating in a tour saw a "browser/chat" window. At the top of this window was a browser, in the middle a scrolling chat dialog box, and at the bottom a line for entering text, either chat messages or URLs. All participants could enter text messages by entering the message and then pressing a "SEND" button, and the message would then be displayed in the dialog box on the screens of all of the tour participants, including the sender.

33. The tour guide could also send URL messages, either by selecting one of the pre-typed URLs from a list, or by typing a URL and then pressing the "URL" button. Upon receipt of the URL message, the browser in the Road Trips window for each participant would automatically fetch the contents indicated by the URL (for example an HTML document), and then display them in the browser window. The contents might be what the plaintiff calls "multimedia messages," including both text and graphical images. Participants in the tour could also operate their browsers independently. For example, if the tour guide sent a URL for a web page (*e.g.*, an HTML document) containing hyperlinks, the user could optionally click on the hyperlink to locate, fetch, and display another "multimedia message" including both text and

graphical images. The user could always return to the URL most recently sent by the guide by pressing the "Last URL" button.

34. Participants in a tour, even if private, could list the members of the tour.

35. Access to the Road Trips software required a test for authentication (the user must first enter a valid AOL member name and password when logging into the system). In addition, AOL's parental controls feature allowed a parent to block access to Road Trips for dependent member accounts created for children. Private tours could only be accessed by entering the secret title of the tour.

36. Road Trips was developed by an AOL employee named Jay Elinsky. He began writing the software for Road Trips on April 25, 1995, or earlier, and had a working version by May 21, 1995. By July 30, 1995, AOL members were using the software. I know these facts from discussions with Mr. Elisnky. He provided me with a printed copy of an HTML document titled "CVS log for manual_tour/src/tour.c" (henceforth "tour.c") (Ex. 4), which indicates when various versions (called "revisions" in this document) of the file tour.c were "committed" by the author. The file tour.c is the main source code file for a program that runs on an AOL server (the controller computer).

37. CVS (Concurrent Versions System) is a well-known and heavily used open-source "version control system." A version control system allows a software developer to save at regular intervals the different versions of the files that make up a program, and allows the developer to view any of the saved versions at any time. A new version (revision) is saved in CVS in a "repository" whenever an author "commits" changes to the software that have been made since the last saved revision. In the related version control system RCS (Revision Control System), changes are saved when the author "checks in" (i.e., commits).

38. Revision 1.1 of tour.c was committed "Tue Apr 25 13:44:01 1995 UTC...by elinsky". UTC (Coordinated Universal Time) is the world-wide standard for time, based on atomic clocks. The author's comment for this revision is "Initial revision." I will discuss the contents of the file tour.c in revisions 1.3 (committed Tue Apr 25 19:00:08 1995 UTC), 1.64 (May 21 18:49:34 1995 UTC), 2.0 (Jul 30 16:20:27 1995 UTC), and 2.1 (Jul 30 17:59:41 1995 UTC) later in this document.

39. When Elinsky began his work on Road Trips, AOL was using another open-source version control system called RCS (Revision Control System). The reason that the document is labeled "CVS log" is that AOL later switched from using RCS to CVS, and converted all of their existing software repositories from RCS to CVS. Such a conversion is not unusual, as CVS uses RCS, but provides many additional features. The dates in the logs were not affected by the conversion.

40. I observed from the log for tour.c that Elinsky worked continuously on the software from April 25, 1995, until August 31, 1995, and then beyond. In particular, the logs (Ex. 4)0020indicate that Elinsky committed changes to the tour.c file nearly every day from April 25, 1995, until June 20, 1995, then sporadically until July 30, 1995, where he once again made changes nearly every day until August 31, 1995. During the gap between June 20, 1995 and July 30, 1995, however, Elinsky had made a copy of tour.c called tour2.c, and began editing that file instead. The log for tour2.c (Ex. 5) indicates that it was created on July 7, 1995, and was changed nearly every day until July 25, 1995. On July 30, 1995, he replaced tour.c with tour2.c, calling the result tour.c version 2.0, and began to edit tour.c once again.

41. Several versions and dates are notable. The first version, 1.1, was created on April 25, 1995 at 13:44:01 UTC. I will discuss version 1.3, committed later on the same day

(April 25, 19:00:08, 1995) below. The log for tour.c indicates, and Jay Elinsky confirms, that the first working version was 1.64, which was committed on May 21, 1995. The first version released on the production AOL system, and used by AOL members was version 2.0. Version 2.0 was committed on July 30, 1995, at 16:20:27 1995 UTC. Some earlier version of Road Trips was released prior to July 30, 2005. I know this because Jay Elinsky has told me there is electronic mail containing feedback from AOL members about the Road Trips service dated earlier.

42. The comments attached to the log entry for version 2.1 indicate "Change tokens from 7 to Y." At any given time, two different versions of Road Trips were installed on the AOL production system. One of these was accessible to members, while the other was used by Elinsky for testing purposes. Each service available on the AOL production system is assigned a unique token number. The two versions of Road Trips were thus assigned tokens "7" and "Y" (which are still reserved for Road Trips today). At different times, the token-number-7 version was accessible to members, while the token-number-Y version was not, and vice versa. When Elinsky was satisfied that a new version, deployed only for testing, was ready for members to access, he would install new forms directing users to the new version (which might have token number either 7 or Y). Tokens denote packets of information that are sent between client software and AOL applications, with the token number specifying the application. In particular, forms send tokens to applications, so the new forms would send tokens with the token number of the new version, rather than with the token number of the old version. Elinsky would then begin using the old token number for testing purposes. The comment in the log entry for version 2.1 indicates that the token-number-7 version (version 2.0 of the source code) has become the version accessible to members (whereas previously the token-number-Y version was

accessible), and subsequently token-number Y would be used for the test version, with development of the test version starting with version 2.1 of the source code. A change in the source code was necessary because the token numbers were "hard coded" in file tour.c. Several lines of code were changed. For example, the C preprocessing directive

```
#define TOKEN_1                    TOKEN_7a
```

in version 2.0, is changed to

```
#define TOKEN_1                    TOKEN_Ya
```

in version 2.1

43. I tested version 2.0 of Road Trips using AOL client software version 2.5 installed with file creation dates of 6/27/1995 or earlier. Hence all of the software, with the exception of a one-line addition to the file tour.c have creation dates prior to 7/30/1995.

44. Mr. Elinsky advised me that all the Road Trips forms were created before July 30, 1995.

45. The functionality of Road Trips can be understood by examining the source code of the server software (for the controller computer) that was stored and data in the repository.

46. The source code for version 1.3 of tour.c, committed on April 25, 1995, indicates that the high-level design for Road Trips, including all of the features claimed in the '491 patent, had already been conceived at this date. There are several notable features in this file:

a. First, the file shows that the software was intended to be executed on an AOL server. This can be seen in the line:

```
#define Q_CONTEXT_LENGTH  320   /* Kludge until the library routines  */
```

b. A Q_CONTEXT is a data construct specific to AOL. Each AOL user had an authenticated user identity. A user could connect to AOL using a variety of communications protocols, including TCP/IP. In addition, AOL

allowed for parental controls, in which case a user's access to chat features on AOL for a parentally controlled account would be restricted.

c. Next, the code indicates that there will be forms with titles such as:

```
#define FORM_WELCOME_TO_TOURS
#define FORM_CREATE_TOUR
#define FORM_ACTIVE_TOURS
#define FORM_TOUR_DESCRIPTION
#define FORM_PEOPLE_ON_TOUR
```

indicating that there would be forms for creating tours, listing active tours, entering a tour description, or listing the participants of a tour.

d. Next comes:

```
/* On FORM_CREATE_TOUR:     */
#define RELID_CREATE_TOUR_TITLE                 1
#define RELID_CREATE_TOUR_DESCRIPTION           2
#define RELID_CREATE_URL_INPUT                  3
#define RELID_CREATE_URL_LIST                   5
#define RELID_CREATE_MEMBTOUR_BUTTON            6
#define RELID_CREATE_PRIVTOUR_BUTTON            7
#define RELID_CREATE_AOLTOUR_BUTTON             8
#define RELID_CREATE_ENDTOUR_BUTTON             9
```

This section indicates that on the form for creating a tour, the users would be able to enter a title and description for the tour, enter URLs and then store them in a list, and then press either the member tour, private tour, or AOL tour buttons to activate the tour. This form also contained the button for ending the tour.

e. The next snippet of code

```
#define RELID_ACTIVE_TOUR_LIST          3
#define RELID_ACTIVE_PRIVATE_TITLE      4
#define RELID_ACTIVE_JOINTOUR_BUTTON    5
```

indicates that a user could list the active tours, enter the title of a private tour, or join a member or AOL tour.

f. The line

```
unsigned  is_tour_guide                :1;
```

in the "per-user structure" indicates that a bit is stored for each user indicating whether the user is the tour guide. Similarly, in the "per-tour structure" there is a field called "tour_guide" indicating who is the guide of the tour.

g. The per-tour structure also has a line

```
char              *current_url;
```

which indicates that each tour will have a current URL, stored as a character string. The current URL is the one that the tour guide has most recently sent to the participants in the tour. The actual button for returning to this URL was labeled "LAST URL".

h. The source code contains lines indicating that the number of users who can simultaneously be a member of a tour is limited:

```
#define     MAX_USERS_PER_TOUR      50    /* But also limited by
users/private room      */
                              /* Eventually may want bigger number for  */
                              /* auditorium-based tours                 */
```

i. These features were further seen and elaborated on in future versions, such as 1.30, in which functional C code was added that, in conjunction with the AOL server software, would perform all of the functions described in the asserted claims. For instance:

i. The "do_execute_url" function in version 1.30 demonstrates what would happen when a URL was received by the controller software:

```
/* Loop through the list of users on the tour, and send the */
/* URL to each one                                          */
for (ndx = 0, count = 0;
       (ndx < MAX_USERS_PER_TOUR) && (count < ptip->num_users_on_tour);
```

```
        ndx++) {
    if (ptip->users_on_tour[ndx] != NULL) {
        count++;
        send_url_to_user (url, ptip->users_on_tour[ndx]);
    }
}
```

ii. Comments in the version 1.30 code demonstrate that the code was intended to start up a web browser on the user's PC and send data to make the browser "fetch" the URL:

```
/* Start up the browser on the user's PC, if not already started, and   */
/* send the atom stream to make the browser fetch the URL               */
```

47. The Road Trips source code, including version 1.3 of tour.c (from April 25, 1995), defines a "tour," which signifies that that the program is intended as communications software. By definition, in any sort of tour, a tour guide would be expected to have a means of communicating with members of a tour in real time. In fact, AOL had such a system in place already for the distribution of messages in real-time.

48. The charts below indicate how the asserted claims of the '491 patent are met by Road Trips. In the chart below, I will refer to the version of Road Trips that I saw in operation (version 2.0). The source code implementing the features of the asserted claims had been written by at least version 1.64 (May 21, 1995).

Claim	'491 Claim Language	Road Trips
1	Computerized human communication arbitrating and distributing system, including:	Road Trips is directed towards such a system (see below).
1(a)	a controller computer;	Different AOL servers perform different functions in Road Trips. One AOL server, for example, authenticates users, while another actually distributes messages, virtually the same way the current AOL Instant Messenger servers work together. To the extent that Road Trips does not meet this particular claim limitation, the current AOL messaging systems do not either. To the extent that multiple computers working in tandem would meet this limitation, Road Trips does as well.
1(b)	a plurality of participator computers	**Participator computer:** The participator computer is a personal computer executing

Claim	'491 Claim Language	Road Trips
		the AOL client software. **A plurality of participator computers:** A plurality of participator computers may simultaneously execute the AOL client software and participate in the same tour, or in multiple tours.
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	The AOL client software produces video output and expects keyboard and optionally mouse input.
1(d)	each said user having a user identity;	Each user has an AOL account name and a screen name. The screen name serves as the user's identity in Road Trips.
1(e)	Connections through the Internet linking the controller computer with each of the participator computers; and	The AOL client software operating on the participator computer allowed the user to connect to the AOL service (i.e., to connect to an AOL server) using TCP/IP over the public Internet, although it uses proprietary protocols controlled by AOL. Nonetheless, under the plaintiff's claim construction, this element is met.
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The controller software consists of the compiled version of the file tour.c and the other AOL host software, which is executed on an AOL server, i.e., the controller computer.
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group:** A "trip" or "tour" is a group. **Group through the controller computer:** The controller computer maintains all information about the group, including the name of the tour, the identity of the tour guide, the members of the tour, the last URL visited, etc. **Plurality of groups through the controller computer:** The controller computer can support multiple tours simultaneously. **Arbitration in accordance with predefined rules including a test for an authenticated user identity:** Only users with "internal" and "overhead" accounts can create "AOL" tours, hence only these users can be guides for AOL tours. (Any AOL users can be the guide for a "member" tour.) A "private" tour can only be accessed by a user who has been given the secret name of the tour. A tour guide can end a tour, ending the participation of all users in the tour. Parental controls can be used to block access to Road Trips for dependent accounts. A tour has a limited number of users, beyond which users

Claim	'491 Claim Language	Road Trips
		are not permitted to enter the tour. In order to access AOL Road Trips, it is first necessary to "log in" to AOL. This requires entering the password for a pre-stored user identity (an AOL member account).
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Messages:** Members of a tour enter messages in a text-entry bar of a "chat room" window provided by the AOL client software operating on the participator computer. The messages are then displayed in the scrolling text portion of the chat room window. **Distributing:** Each message is sent from a participator computer to the controller computer which then distributes the message to the participator computers belonging to all the members of the tour, including the sender. This behavior is demonstrated in the tour.c code, versions 1.30 and 2.0. **Real time:** Messages appear in the scrolling text portions of the chat room windows immediately after they are sent.
1(i)	at least some of the user messages are multimedia messages.	A user may send a URL in a message. The tour guide may also send a special "URL" message by, for instance, clicking the "URL" button rather than the send button. In the latter case, a specially tagged URL message is distributed to the group, the URL is recognized as pointing to a web page which may contain multimedia content, and the web page is automatically displayed on all of the screens of the users on that tour.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The participator software is the AOL client software, which operates on the participator computer. This software provides a chat window that permits a user to send messages to a controller computer, thus enabling it to arbitrate and distribute the messages.
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	The tour guide may send a URL message, which compels the participator computers to locate a second message (a web page). This web page may contain links to other web pages. A user may then click on a link on the second multimedia message to optionally locate another multimedia message, e.g., a third message consisting of a web page containing both text and graphics. Alternatively, any member of a tour may send a message containing a URL, which the users on the participator computers may then use to optionally locate another

Claim	'491 Claim Language	Road Trips
		multimedia message by copying the URL using the Windows copy function, and then pasting the URL into the address bar of a browser using the Windows paste function.
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	The tour guide may send a URL message, which compels the participator computers to locate a web page, which might be a multimedia web page containing text and graphics, and hence a multimedia message, and then display it in a browser.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	The browser is contained in a separate window or "subscreen" on the output device by the AOL client software.
6	The system of claim 4, wherein the other message is a multimedia message.	See claim 4.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	America Online stores a variety of information about each member, including the user's age (for parental controls), email address, name, etc.
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	When a member joins a tour, the controller computer indicates to the participator computer that it is to invoke the AOL browser, IWENG.DLL, which is a Microsoft Windows dynamic link library, i.e., a separate computer program. Multimedia messages, such as web pages are displayed in the browser upon their receipt.

Claim	'491 Claim Language	Road Trips
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	See claim 26.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective	See claim 1.

Claim	'49' Claim Language	Road Trips
	ones of the participator computers,	
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controlier computer, the authenticated user identity including respective representations of at least one member from the group consisting of age, telephone number, fax number, name, , company, postal address, E-mail address, and URL.	See claim 8.
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controlier computer, the authenticated user identity including respective representations of at least three members from the group	See claim 8.

Claim	'491 Claim Language	Road Trips
	consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

49. If called to testify at trial, I will be prepared to demonstrate one or more versions of AOL Road Trips, including version 2.0, running in conjunction with one or more versions of the AOL client software version 2.5. I will also be prepared to exhibit and testify about the code for tour.c versions 1.3, 1.30, and 2.0. I will also be prepared to make demonstrative exhibits from the above, such as by using screenshots, representations of screenshots, or other exhibits to demonstrate my opinions.

50. The Netscape Chat version 1.0 software meets all of the asserted claims in the '491 patent.

51. Netscape Chat version 1.0 is a chat program that also allows users to send URLs to other users. The web pages corresponding to these URLs can then be viewed either automatically or optionally in Netscape Navigator, version 1.22, depending on the chat user's preferences (the Auto View option). These two programs operate on the chat user's personal computer, which serves as the participator computer. Hence Netscape Chat is participator software. Netscape Chat connects over the Internet using TCP/IP to another computer, the controller computer, which is running an "Internet Relay Chat" (IRC) chat server. The IRC chat server is the controller software. IRC is an open protocol, first defined in RFC 1459, in May 1993. There are many implementations of IRC servers, and open-source implementations existed at least as early as March 8, 1995.

52. Netscape Chat, in conjunction with an IRC server, offers a variety of arbitration and authentication options. For example, a channel operator may kick another user out of a channel, or ban a user from joining the channel based on his identity (a combination of a nickname and a user name). Netscape Chat supports authentication when used in conjunction with an IRC server that has been configured to store authenticated user identities. In particular, when Netscape Chat connects to an IRC server, it sends a user name and password to the server. The server then determines if the user name has been registered and whether the password is correct before allowing the user to participate.

53. I installed Netscape Chat using Netscape's Power Pack CD-ROM which automatically runs the program powerpack.exe. On this CD-ROM, the creation date for the file Powerpack.exe is 10/3/1995. The executable file for version 1.0.1.8 (the 32-bit version) of

Netscape Chat is called nschat.exe, and has a creation date of 9/18/1995. I was not able to determine the date for version 1.01 (the 16-bit version) of the Netscape Chat program file NSCHAT.EXE. It is installed with a creation date of 01/01/1980, which I believe is a default creation date. (Its true creation date must precede that of powerpack.exe, however.)

54. I obtained the Undernet IRC chat server, ircd, from ftp.undernet.org. In particular, I retrieved a file titled ircu2.9.19.tar.gz, which is a compressed archive of source code files. The latest file creation date in this archive is March 8, 1995.

55. I reviewed and evaluated the Netscape Chat source code. The source code files contain CVS date information which dates the code as early as August 11, 1995. The code dated as of August 11 appears to be fully functional and contains all of the features of the asserted claims when used in conjunction with an IRC server. If that code, or code containing the same functionality, had been compiled and tested, that testing would establish reduction to practice of all of the features of the asserted claims.

56. I also reviewed design documents for the Netscape Community System, of which Netscape Chat appears to have been a part. Specifically, I reviewed four HTML files which, together, demonstrate conception of an IRC-based system with additional "multi-media message' capability.

a. The document entitled "Feature List for Release 0.8", dated May 10, 1995, discusses a feature list for a chat server which conforms to IRC RFC 1459 (the primary IRC specification, attached as Ex. 11. It mentions that the system has "Support for One to One" or "Group Conferencing (basic IRC chat channel)" in which "multi-media messages" are exchanged. "The

multi-media data could be anything (audio, url, image etc.) [Type of multi-media data is client's issue]."

b. A document entitled "The Community Project," dated May 5, 1995, refers to "Multi-Media Chat" and "Netscape Chat" interchangeably.

c. The document entitled "Multi-media Chat Protocol – Message Format", dated April 21, 1995, explains the idea of using an IRC-compliant client/server and adding additional data to the message for multimedia data that a normal IRC client would not process.

d. The document entitled "Chat Objects," dated May 5, 1995, discusses user information for a chat system, including "real name, nick name, and other personal information."

e. These documents collectively demonstrate that as of April 21, 1995, Netscape had conceived of using an IRC system to support the sending of multimedia content, including URLs, over IRC channels. This is precisely what Netscape Chat does. Thus, this system, to the extent it supports URL messages, was reduced to practice with Netscape Chat.

57. The charts below indicate how the asserted claims of the patent are met by Netscape Chat.

Claim	'491 Claim Language	Netscape Chat
1	Computerized human communication arbitrating and distributing system, including:	Netscape Chat is directed towards such a system (see below).
1(a)	a controller computer;	The controller computer is the computer operating the IRC server software.
1(b)	a plurality of participator computers	**Participator computer:** The participator computer is the personal computer operating the Netscape Chat software.

Claim	'491 Claim Language	Netscape Chat
		A plurality of participator computers: Multiple participator computers may operate the Netscape Chat software, and multiple participants may join one or more IRC channels.
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	The Netscape Chat software produces output for a video display, and accepts input from a keyboard and optionally a mouse.
1(d)	each said user having a user identity;	Netscape Chat distinguishes three types of names: nicknames, user names, and real names. The nickname is the name that is used to identify a user to other chat members. It serves as the user identity. The user name is provided to IRC servers so that those that limit access to registered users can determine if the user is registered A user of Netscape Chat optionally enters a real name. If a real name is entered, other IRC users are able to see it.
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	Netscape Chat, operating on the participator computers, connects over the Internet to an IRC server operating on the controller computer using the TCP/IP protocol.
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The controller software is the IRC server software, ircd.
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group:** An IRC channel is a group. **Group through the controller computer:** All of the information about an IRC channel, including the channel name and the list of members, is maintained by the IRC server operating on the controller computer. **Plurality of groups through the controller computer:** An IRC server operating on the controller computer can support a plurality of channels (groups). **Arbitration:** Some IRC servers limit access to registered users. In addition, through the MODE command, Netscape Chat in conjunction with an IRC server provides a large number of arbitration options, including (from RFC 1459 and the help documentation provided with Netscape Chat) the following modes: o - give/take channel operator privileges; p - private channel flag; s - secret channel flag;

Claim	'491 Claim Language	Netscape Chat

		i - invite-only channel flag;
		t - topic settable by channel operator only flag;
		n - no messages to channel from clients on the outside;
		m - moderated channel;
		l - set the user limit to channel;
		b - set a ban mask to keep users out;
		
		To exercise one of these modes, a Netscape Chat user would type a special command starting with "/mode". As an example, to ban a user with nickname PeteWork from a channel called #AOL1, a Netscape Chat user starts by typing the following message:
		`/userhost PeteWork`
		
		The IRC server would then respond with a message such as
		`PeteWork=+~peter99@66.28.38.176`
		
		The user would then type
		`/mode #AOL1 +b PeteWork!*peter99@66.28.38.176`
		
		User PeteWork would then be banned from the channel. That user cannot re-enter the channel even if he logs out of the system and logs back in with a different nickname.
		
		Alternately, PeteWork could be kicked out of the channel, but not banned, by entering the command
		`/kick #AOL1 PeteWork`
		
		In accordance with predefined rules including a test for an authenticated user identity:
		
		According to RFC 1459, an IRC server can authenticate users in one of two ways. First, an IRC server may employ a global password that must be provided before a user can connect to the server. Second, an IRC server may store individual passwords for registered users. In this case, a user must present both a valid registered user name and the password associated with that name.
		
		Netscape Chat passes both user names and passwords to IRC servers.

1(h)	Distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing:** By default, all messages are first sent by participator computers to the controller computer over the Internet using TCP/IP. The controller computer then distributes the messages to the participator computers over the Internet, for example to all of the participator computers in a channel (group). **Real time:** IRC messages are delivered in real time.

Claim	'491 Claim Language	Netscape Chat

1(i)	at least some of the user messages are multimedia messages.	Netscape Chat allows a user to send a URL to the other users in the same channel. The user does this by entering the URL in a text entry bar separate from and beneath the chat entry bar, and then presses the Send button. The message is identified as a special message. For example, a normal text message might appear as:
		
		<Bruce> hello!
		
		Whereas a URL would appear as
		
		<Bruce shows> http://www.aol.com
		
		The corresponding web page is then automatically located and displayed by the Netscape Navigator web browser provided that the Auto View option is on, which is the default setting.
		
		The source code file "ncapp2.cpp" also confirms this functionality. In the following excerpt, when a message is received from a chat server, the client is aware of whether or not the message "IsURL()", i.e., is a URL, and if so, appends " shows" to the name of the sender.
		
		`//////////////////////////////////////////////////////////////////////`
		`// message receive from a IRC channel`
		`//////////////////////////////////////////////////////////////////////`
		`case ieChannelMsg:`
		`{`
		`  // I got a message from Chat server,`
		`  // someone is talking to me.`
		`...`
		`  BOOL  isurl = msgirc->IsURL();`
		`  CString  msg = isurl ? " shows" : _T("");`
		`  pDoc->processChatData( isurl, sender, body, msg);`
		`  }`
		`}`
		`break;`

2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The Netscape Chat software is the participator software operating on the participator computers and enabling users to send messages to the controller computer (the IRC server) over the Internet using TCP/IP, and thus enables the arbitrating and the distributing of user messages by the controller computer.
3	The system of claim 1, wherein: the user messages include an address to instruct the	A user message may be a URL, which contains the address to optionally locate another multimedia message such as a multimedia web page (e.g., one that contains both text and

Claim	'291 Claim Language	Netscape Chat
	participator computers to optionally locate another multimedia message.	graphical images). The sender of the message can either type the URL and then press the Send button, or, if the "Auto Send" option in Netscape Chat is on, any link clicked on in Netscape Navigator is automatically sent to the other users in the same channel. If the receiver has turned off the Browser | Auto View option in Netscape Chat, then the other multimedia message (a web page containing text and graphics) can optionally be located by clicking on the URL in the list of recently received URLs. The message is then displayed in the Netscape Navigator browser.
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	By default, the participator computer is compelled to locate another message. In particular, if the Auto View option is on (the default setting) , when a URL message is received, the Netscape Chat program is compelled to instruct the Netscape Navigator browser to locate the message and present it at an output device.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	The Netscape Navigator browser operates in a different window (subscreen) than the Netscape Chat program.
6	The system of claim 4, wherein the other message is a multimedia message.	The other message may be a multimedia web page containing text and graphics.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	Netscape Chat allows a user to enter a username, nickname, and real name and sends that information to the IRC server using the "/user" command. A user could easily append his e-mail address, phone number, and any other information he wished to his real name field. Additionally, Netscape had conceived of storing "real name, nick name, and other personal information," in a design document named "Chat Objects," dated May 5, 1995. To the extent all of these pieces of identifying information were not used in Netscape Chat, at the very least, it would be trivial to one of skill in the art to add additional pieces of identifying information, and the motivation to do so is clearly present in the "Chat Objects" document.
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator	The Netscape Chat program operating on the participator computer locates the Netscape Navigator browser and invokes it. The browser then displays multimedia messages on the respective output device. This functionality is confirmed in the "processChatData" function in "nc3doc.cpp", which is called when a message is received, and recognizes when a message is a URL and, if so, "pushes" it to the browser:

Claim	'491 Claim Language	Netscape Chat

	computers; and
	
	invoking the computer program to present the multimedia message at the respective output device.
		`if( isurl )`
		`{`
		`  if( m_urlEnabled )`
		`  {`
		`...`
		`  if( m_urlAutoViewFlag )`
		`  {`
		`  // pump it to the browser`
		`  sendUrlToBrowser( CString( body ) );`
		`  }`
		`  } // if( m_urlEnabled )`
		`} // if( isurl )`

27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	See claim 26.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.

Claim	'491 Claim Language	Netscape Chat
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one member from the group	See claim 8.

Claim	'491 Claim Language	Netscape Chat
	consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

58. If called to testify at trial, I will be prepared to demonstrate the Netscape Chat software in conjunction with an IRC server. I will also be prepared to testify about the Netscape Chat source code and design documents, as well as the IRC standard based on RFC 1459 and other sources. I will also be prepared to make demonstrative exhibits from the above, such as by using screenshots, representations of screenshots, or other exhibits to demonstrate my opinions.

59. CompuServe Producer is a program that allows one user (the producer) in a chat group to send audio recordings and video snapshots to the other users in the chat group. The Producer software I used was called "CompuServe Producer" V.198C, Copyright 1996. In particular, I executed a program called csprod.exe that had a file creation date of 5/31/96. The producer software existed in a substantially identical form by at least December of 1995, as evidenced by the video attached as Ex. 29. Other participants in the chat room used the "CompuServe Information Manager," running the program WINCIM.EXE, which has a file creation date of 10/31/95 (and which is also shown in the video attached as . The other participants could receive an audio stream and see video snapshots, but could not send audio or video.

60. In analyzing CompuServe Producer, I examined source code for CompuServe's Producer and Viewer, and also for CompuServe's "CB Conferencing" system, which ran on CompuServe servers.

61. WINCIM.EXE launches executable code that implements CompuServe's "Viewer," which is used by a participator computer to send and receive messages, including text, video and audio messages sent by Producer through a CompuServe server. The document "CompuServer Producer Station System Requirements" indicates that "the Viewer software is included in WinCIM 2.0.1 and higher."

62. Several dates are documented in the files provided as Ex. 13. The following, for example, quoted from file AOL0071887(CompServe_Producer_Viewer_code/ PRDOVR-May-95.txt, indicates that Viewer was ready for production at least as early as May, 1995, meaning that it was ready to be included as part of the standard CompuServe client software, and that it was demonstrated to journalists at that time.

Research and Development Monthly Overview

May 1995

Jeffrey S. Miller

CompuServe Viewer

Moving to production! The CompuServe Viewer was released to the Columbus IPG group to get it into production. The version released does include SLAP, which enables the Viewer to be launched from within WinCIM. Adding SLAP took a fair amount of effort and I'd like to thank everyone involved for their effort in making it work. Jeff Dalton is documenting the steps involved in doing a successful SLAP. If you want a copy just ask. Some last minute user interface changes were added to the Viewer and the latest release is available on the INSIDE forum (Version 1.70).

The Producer component was also given to IPG, however we'll continue to do development in this area. Also, full Windows installation systems were created for the Viewer and the Producer.

CNN has been demoing the Viewer to journalists and other companies. Longer term is to do more with CNN and better handling of closed caption (and other information) with relation to the audio and video.

63. The source code files contain comments indicating the dates on which various changes were checked in, and the effect of those changes. As an example, the file "/AOL0055618(CompServe_CB_Conferencing_Code/server/conf.cxx," software that runs on a CompuServe server, indicates that Revision 1.1 was checked in on 1993/05/27, and that Revision 1.59 (the last listed) was checked in on 1995/10/04, with continuous development in-between.

Similarly, file "/AOL0052114(CompServe_Producer_Viewer_code/CS-Producer/Prodmain.c", indicates that the initial version of Producer (Revision 001) was developed by 12-16-94, with Revisions 002, 003, 004, and 005, being completed by 5-30-95, 10-30-95, 12-01-05, and 12-21-95, respectively. Finally, file "AOL0052114(CompServe_Producer_Viewer_code/CS-Viewer-Sources/CSVIEWER.C" shows Revisions 001, 002, and 003 of the Viewer software occurring on 12-05-94, 04-06-95, and 10-03-95, respectively.

64. The CompuServe Producer / Viewer system contains or suggests all the limitations of several of the Claims at issue in this case, as shown below.

Claim	'491 Claim Language	CompuServe Producer / Viewer
1	Computerized human communication arbitrating and distributing system, including:	CompuServe Producer / Viewer is directed towards such as system (see below).
1(a)	a controller computer;	Different CompuServe servers performed different functions in this system. One CompuServe server, for example, authenticates users, while another (operating the CompuServe CB Conferencing system software) actually distributes messages. Windy City alleges that such multiple servers meet the claim. In any event, it would have been obvious to one skilled in the art to use a single server (for instance, for a smaller-scale installation).
1(b)	a plurality of participator computers	**Participator computer:** There are two types of participator computers. One participator computer runs the CompuServe Producer software. The other participator computers are those that run CompuServe's CIM software, which includes Viewer. **A plurality of participator computers:** A plurality of participator computers may run the CompuServe CIM software.
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	Both the CompuServe Producer software and the CompuServe CIM software produce a display for a video monitor, and accept input from a keyboard and mouse.
1(d)	each said user having a user identity;	The CompuServe user ID is the user identity.
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	The participator computers running the CIM software could connect to the controller computer (the CompuServe server) using TCP/IP.

Claim	'491 Claim Language	CompuServe Producer / Viewer
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The controller software is the CB Conferencing System software operating on the CompuServe server (the controller computer) that manages the chat session, receives messages from the participants and audio and video from the producer, and distributes these messages and the audio and video among the participants.

1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group:**
		A group is a room in a particular service or a private group.
		Group through the controller computer:
		All information about groups is stored in the controller computer, including the list of members, the name of the group, etc.
		Plurality of groups through the controller computer:
		The controller computer can support a plurality of groups. In my review, there were at least 9 different rooms in the "CATHOLIC" service.
		Arbitration in accordance with predefined rules including a test for an authenticated user identity:
		The user must provide a valid CompuServe user ID along with the corresponding password before gaining access to the service.
		The CB Conferencing system software indicates that a user could create a group and then invite other users to join the group. Without receiving an invitation, a user would not know the number of the group to join. If this is deemed "arbitration," then this is an additional form of arbitration.
		In particular, file "AOL0055618(CompServe_CB_Conferencing_Code/server /group.cxx" (Ex. 13) observes that as of Revision 1.24, dated 1993/10/19,
		`// 11 Invitations are now associated not only with the invitign user,`
		`//  but also with the group from which the invitation was issued.`
		`//  this prevents a user from inadvertantly /JOINing a group to`
		`//  which he was not invited.`
		The file group.cxx implements the following functions.
		`/*`
		`<f><s> Make Group`
		`This function processes the CCP_MakeGroup event`
		`After identifying the user, This function attempts to allocate a group resource`
		`and construct a conferencing object for it. The resulting group number is`
		`returned to the user as acknowledgement.`

Claim	'491 Claim Language	CompuServe Producer / Viewer
		/ int Make_Group(int conidx, msgb msg) ... /* <f><s> Add To Group This function processes the CCP_AddToGroup event from the client The function verifies the existance and reacability of each user, and send a CCP_Invitation event to them. */ int Add_To_Group(int conidx, msgb* msg) ... /* <f><s> Invitation This function handles a CCP_Invitation Event from a remote Server. The event is translated into its local form, and forwarded to the target user */ int Invitation(int conidx, msgb* msg) ... /* <f><s> Join Group This function handles the CCP_JoinGroup Event */ int Join_Group(int conidx, msgb* msg)
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing:** The controller computer distributes text, audio, and video to the respective participator computers, i.e., to the participants in the same room. Messages from Producer are distributed to participator computers through the controller computer. For instance, all messages from Producer must pass through the controller computer before reaching participator computers. The C++ program file for CompuServe's CB Conferencing system software, which operates on a CompuServe server, AOL0055618(CompServe_CB_Conferencing_Code/server/conf.cxx (Ex. 13) defines a class called "Conference". The following quote from conf.cxx indicates that a room is a type

Claim	'491 Claim Language	CompuServe Producer / Viewer

		of conference:
		`// Revision 1.14 1994/07/07 12:07:48 rambrose`
		`// (RWA) Revised the CCP Protocol with respect to the`
		`// addressing of Conference objects (Rooms and groups).`
		`// All objects of type Conference have a system wide`
		`// unique address of type ConfRef (see common/ccp.h for`
		`// the definition). This address is now used`
		`// exclusively to identify Conference objects which are`
		`// the source or destination of a CCP message.`
		`//`
		`// . The handling of all CCP Messages which now use`
		`// ConfRef addressing has been altered to search for the`
		`// destination using the ESearch() function (defined in`
		`// server/conf.h) to search the entire list of`
		`// Conference objects. The net effect is to allow any`
		`// Conference object, whether it is a room (public) or a`
		`// group (private) to be addressed by any message`
		`// affecting Conference objects.`
		
		In the same file, the following code snippet demonstrates that the server distributes a message sent to a room to the respective ones of the participator computers.
		`int Conference::Sendto(msgb* msg)`
		`{`
		`ConfMember *c;`
		`unsigned int i;`
		`int j;`
		
		`if (member.Count() == 0)`
		`return 0;`
		
		`for (j=i=0, c=member.Peek(); i<member.Count(); i++, c = member.Next())`
		`{`
		`if (c->isValid())`
		`j += c->user->Sendto(msg);`
		`else`
		`{`
		`member.extract();`
		`speakers -= c->Speaker;`
		`delete c;`
		`}`
		
		`}`
		
		`return j;`
		`}`
		
		Real time:
		
		Messages are delivered with little perceptible delay after they have been sent.

Claim	'491 Claim Language	CompuServe Producer / Viewer
1(i)	at least some of the user messages are multimedia messages.	The Producer software indicated a feature in which multimedia messages could be sent by combining a video stream with generated closed captions. A video snapshot with text from a closed caption is a multimedia message. A user may send a URL to the participator computers in a group. One of the users may copy the URL and paste it into the address bar of a web browser to display the multimedia contents of the corresponding web page. The system supports multimedia messages via the sending of text by the Producer software at the same time that an audio file sent by the Producer software is played in the CIM software. Furthermore, the ability to send messages containing tagged URLs which are recognized by the client as such and to which the client is responsive, would be obvious to one of skill in the art, as this feature was contemporaneously implemented by Road Trips, WebTalk, and other software, in conjunction with browsers such as Mosaic or Netscape Navigator.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The participator computers ran the CIM software. Each of these programs allows a user to send a message to the controller computer (the CompuServe server hosting the service). Upon receipt of these messages, the controller computer would then be enabled to distribute the messages back to the appropriate participator computers. The participator software would then receive the messages and display them. The file CSVIEWER.C, in the archive "AOL0052114(CompuServe_Producer_Viewer_code)\CS-Viewer-Sources.zip," (Ex. 13), for example, implements functions such as "DisplayImage", "DisplayText", "PlaySound", and "MsgCreate". Similarly, the file "Prodmain.c" in the archive "AOL0052114(CompServe_Producer_Viewer_code\CS-Producer.zip" *(Ex. 13), implements functions such as "CSPSendClosedCaption", "DisplayText", "MsgCreate", and "GrabFrame".
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	A user could type a URL into the text entry form. Upon receipt, another user could optionally copy the URL using the Windows Clipboard into the address bar of a browser, which would locate and display the corresponding web page, which might contain text and graphics, and hence be a multimedia message. This page might also contain a link that the user could then optionally click on to bring up another multimedia message. Also, it would have been obvious to include a clickable link. See claim 1.

Claim	'491 Claim Language	CompuServe Producer / Viewer
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	Under the plaintiff's claim construction, which does not limit the act of compelling to preclude user action, this claim limitation would be met by a user, receiving a URL in a text message, copying that URL into a web browser and forcing the browser to display the web page.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	See claim 4.
6	The system of claim 4, wherein the other message is a multimedia message.	See claim 4. The web page displayed can contain text and graphics, as well as links to other web pages with multimedia content.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	CompuServe stored a variety of information for each user identity, including name, e-mail address, and postal address.
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	To locate and invoke a separate computer program, such as a web browser, to process a URL that might be included in a user message, would have been obvious to one of skill in the art, since Mosaic and other web browsers had long provided the functionality of locating and invoking "helper" programs to process different types of data.
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	See claim 26.

Claim	'49' Claim Language	CompuServe Producer / Viewer
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.

Claim	'491 Claim Language	CompuServe Producer / Viewer
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one member from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.

Claim	'491 Claim Language	CompuServe Producer / Viewer
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

65. If called to testify at trial, I would be prepared to demonstrate the Producer system, and the Compuserve CB Conferencing System, in conjunction with the WinCIM software and supporting hardware and software, and source code. I will also be prepared to make demonstrative exhibits therefrom, such as by using screenshots, representations of screenshots, or other exhibits to demonstrate my opinions.

66. Gtalk is a chat system written by David W. Jeske and Daniel Marks (the inventor named in the '491 patent). Marks and Jeske prepared versions of Gtalk for several operating systems including DOS, OS/2, and UNIX. I obtained the source code for these versions from

David Jeske's web site http://mozart.chat.net/~jeske/Gtalk/. The creation dates for the DOS19 and DOS19z14 source code files (.C and .H) are all prior to 1994. The creation dates for the OS2 source code files are all March 20, 1995, or earlier. The source code files for UNIX versions new1.6.4 and v1.6.4r2 contain copyright dates of either 1993 or 1995. Finally, the Gtalk Owners Manual is dated July 14, 1995.

67. Gtalk's functionality met all of the limitations of the asserted claims of the '491 patent.

68. In addition to allowing users to send text, Gtalk allows users to send characters from the high ASCII character set. In particular, there was a flag called "HIGH_ASCII_TOG" indicating whether these characters were allowed. Several sample lines of code involving HIGH_ASCII_TOG are shown below. The first is from "toggles.h":

```
#define HIGH_ASCII_TOG              11
```

and the others from gt.c:

```
if  (line_status[portnum].ansi)
{
  if (test_bit(user_options[portnum].toggles,HIGH_ASCII_TOG))
    line_status[portnum].ansi |= 0x02;
}
```

69. Some of the high ASCII characters are images rather than text. In particular, characters 176 through 227 (decimal) on an IBM compatible PC (e.g., one running DOS), which uses the IBM Extended ASCII Character Set, are graphical images. These characters are shown below:

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	▒	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

70. According to the Gtalk manual (p.6):

1.3.7: Gtalk Extended Characters

The IBM Extended Character set is handled in a similar manner. The set of |+xx codes is for allowing extended ASCII characters. If the user does not have Extended ASCII enabled then he will see the normal ASCII character which closely resembles the Extended ASCII character.

71. When the extended character set is enabled, graphical ASCII characters can be sent using the following format:

|+xx

where "xx" denotes a 2-digit hexadecimal (base 16) number. Graphical ASCII characters exist in the numeric range from 176 (B0) to 223 (DF). For example:

```
|+CC
```

adds the "╠" character (204, or CC in hex) to the typed message. The user can type normal text characters as part of this string as well.

72. A message containing both text characters and graphical image characters is a multimedia message.

73. At least the UNIX version of Gtalk, 1.6.4, and possibly other versions, also allowed a user to send a "beep" message to another user by typing Control-g in a message along with other text. For example, if a user typed:

```
Hi there.  Here's a beep.
```

followed by the keystrokes "Ctrl" and "g" simultaneously, and hit "Enter," the message

```
Hi there.  Here's a beep.
```

would be transmitted to all group members, and an audible beep would simultaneously be heard by all participator computers in the group.

74. The OS/2 version of Gtalk, as well as the UNIX versions "beep the console" on occasion. For example, the UNIX version 1.6.4 beeps when a user first enters Gtalk, and again after the user logs in. Users of the OS/2 version are also able to send beeps, in conjunction with text messages through the user of the /PAGE command. The code that does this is shown below.

```
print_string("--> Paging.");
for (loop = 0;loop<10;loop++)
  {
    print_chr_to(7,node);
    print_chr('.');
  }
print_str_cr(".Done");
sprintf(s,"--> Paged by %c%s|*r1%c",user_options[portnum].staple[2],

user_lines[portnum].user_info.handle,user_options[portnum].staple[3]);
aput_into_buffer(node,s,0,8,tswitch,node,3);
```

On the pager's terminal, this would print

```
Paging.....Done
```

75. As each dot (".") except the first and last printed, a beep character was sent to the paged user. The beeps were sent rapidly, so that the pager would hear ten beeps in very quick succession. The beeps would be accompanied by a text message that appeared on the user's screen indicating that he/she had been "Paged by" the pager.

76. A message containing text characters and an audible beep is a multimedia message.

77. Hence, Gtalk allows the sending of multimedia messages either through the use of graphical image characters or audible beeps, in conjunction with text characters.

78. As Marks testified, the Unix version of Gtalk (at least versions 1.6.4 and later) supported a feature called "Game Connection" (or "GAMECON", for short). GAMECON allowed users of the multiplayer game DOOM (the DOS operating system version of DOOM) to form a "virtual" IPX network despite the fact that they were not located on the same physical local area network. In particular, each computer on which the DOOM program was operating would make a serial connection to Gtalk (e.g., using a modem) and the users would then join the same Gtalk channel. The IPX packets generated by DOOM would be "tunneled" over the serial connection to Gtalk, which would then redistribute the packets to the computers of the users in the same channel. A DOOM packet could contain a chat message as well as information specifying the movement of the DOOM character representing the sender of the message. DOOM would display the chat message on the screen of the recipient, and also simultaneously update the multimedia display depicting the location of the DOOM characters. Such a message, consisting of both text and a new graphical display, is a multimedia message.

79. The motivation for GAMECON was that IPX was a proprietary protocol of Novell, and is not the same protocol that is used on the Internet (TCP/IP). DOS users of DOOM

whose computers were not on the same local area network could not, therefore, play the game together.

80. Although GAMECON was designed to allow DOS users of DOOM to connect to Gtalk using serial connections, it would also be straightforward to allow DOS users to connect to Gtalk using a TCP/IP connection. Indeed, the Unix version of Gtalk also supported TCP/IP connections, and it would be trivial for one skilled in the art to make the changes (if any) to GAMECON needed to support TCP/IP connections. Furthermore, the changes would be well-motivated, as there was a clear desire to allow players of the DOS version of DOOM at diverse locations to play against one another, and indeed GAMECON was designed for this purpose.

81. The following charts indicate that Gtalk contains all of the limitations of the claims at issue in this case. Quotes are taken from the Gtalk Owners Manual. Any comments not in quotes indicate material learned from inspecting the Gtalk software and testing it.

Claim	'491 Claim Language	Gtalk
1	Computerized human communication arbitrating and distributing system, including:	Gtalk is directed towards such a system.
1(a)	a controller computer;	The controller computer is the computer on which the Gtalk software operates.
1(b)	a plurality of participator computers	**Participator computer:** The participator computer is the computer on which the telnet software or terminal emulation software operates. **A plurality of participator computers:** A plurality of participator computers may each run the telnet or terminal emulation software.
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	The telnet and terminal emulation software operating on the participator computer each expects to receive input from a keyboard and produce output on a video screen and a computer audio speaker.
1(d)	each said user having a user identity;	A user has a numeric user id in Gtalk.
1(e)	connections through the Internet linking the controller computer with each of the participator	In the Unix version of Gtalk, participator computers can use the telnet application for connections to the controller

Claim	'720 Claim Language	Gtalk
	computers; and	computer through the Internet using the TCP/IP protocol. In the DOS and OS/2 versions, participator computers make serial connections to the controller computer via modem. Marks testified that he used the Internet to connect to a computer attached to a modem bank and then to gtalk, thereby connecting a participator computer to the controller computer through the Internet.
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The controller software is the Gtalk software.
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group:** A group is called a "channel" in Gtalk. Participants belong to channels. "2.2 Channels Channels allow people at the Main Interaction Level to collect into sub-groups to have conversations." **Group through the controller computer:** The controller computer maintains all of the information about each channel, including the name of the group and which members belong to which channels. **Plurality of groups through the controller computer:** The Gtalk software supported a plurality of groups on a single controller computer. "A.3 Channels There are several "channels," that users select when they want to split up into different groups and talk." **Arbitration in accordance with predefined rules including a test for an authenticated user identity:** Gtalk provides a wide variety of arbitration mechanisms. For example, one participant (a channel moderator) can ban another participant from participating in a channel. A channel can also be made open only to invited participants, and the channel moderator can invite and uninvite participants. "2.2.1 Channel Moderators Channels are controlled by channel moderators... Channel moderators have many abilities to control a channel. These are highlighted here. Channel Locking A channel may be locked with the /CL+,- command. When a channel is locked only those who are on the channel invite list , to be described later, will be allowed on the channel. Channels may also be locked by priority by the /CP command. When a channel is priority locked, only

Claim	'491 Claim Language	Gtalk
		those users who have a priority lower than or equal to the channel lock priority (or who are invited to the channel) will be allowed to enter the channel. Channel Invite List Users may be added to or removed from the channel invite list with the /CI command. Any user may be invited to any channel. Removing Users from a Channel A user who is becoming a nuisance or who is otherwise not wanted on a channel may be removed from the channel by one of the channel moderators. The /CK command allows this. A user who is kicked from a channel will arrive on that user's login channel. A user cannot be kicked from his login channel." "If a valid login ID and password are entered then the user will start logging into the system."
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing:** All Gtalk software ran on the controller computer. There were two components, a "server" component and a "client" component, and one instance of the client is run for each participant. The client component would accept typewritten messages from the corresponding participant, and then pass them to the server component. The server component would then distribute the messages among the intended client components, which would then deliver the messages to the respective participator computers. "2.1.10 Normal Messages to a Channel Most system interaction between users is through normal "spoken" messages. Any text which is not prefaced by one of the system command characters is interpreted as a normal message. [Appendix B] A normal message is printed to all users who are currently viewing the main channel of the user who typed the message. It is prefaced by information about the user who typed the message. (see figure sample)" **Real time:** Messages were delivered in Gtalk in "real time", i.e., with minimal appreciable delay between the time a message was sent and the time it was received.
1(i)	at least some of the user messages are multimedia messages.	Gtalk supports sending two types of multimedia messages: messages combining text characters and graphical image characters, and messages combining text characters and audible beeps. "2.1.1 Choosing Your Terminal Type A menu will be presented after a connection is made... [F]ull Screen Ansi

Claim	'491 Claim Language	Gtalk
		This option will provide the user with an interface which utilizes ANSI color, ANSI screen positioning, and the IBM Extended Character Set." Gtalk also allowed sending an audible beep along side a text message through use of the Control + G keystroke combination, as discussed above. Gtalk allowed users to "page" each other directly by sending an audible beep simultaneous with a private message. Under the plaintiff's claim construction of claim 1, this form of private messaging meets the limitations of claim 1, including arbitrating into a plurality of groups and distributing the user messages, in addition to "multimedia messages." Finally, according to the inventor, Mr. Marks, Gtalk had a function called GAMECON", which provided the ability to distribute messages between participator computers on which the DOOM video game was being run. DOOM, supported animated graphics and sounds. Users could use Gtalk to exchange text messages chat with each other while playing the DOOM game. This functionality meets the limitation of "multimedia messages" as well. Although GAMECON only natively supported the IPX communication protocol (as opposed to TCP/IP), it would have been obvious to construct a system that worked with TCP/IP, since Gtalk already provided TCP/IP connectivity
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The telnet software or the terminal emulation software operating on the participator computer is the participator software. This software enables the users to send messages to the controller computer, which is then enabled to arbitrate and distribute messages.
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	A user can type a URL and send it as a message. Upon receipt of such a URL, another user can then optionally copy the URL into the address bar of a browser and locate and display the corresponding web page. The web page might contain both text and images, making it another multimedia message. Additionally, in order to indicate that the user is sending a special URL message, he has the capability to change his username from "name" to "URL from name." Also, while using DOOM in conjunction with Gtalk's GAMECON feature, a user could sent a message to another user indicating the address or location within the DOOM virtual world at which a specific feature, such as a treasure or monster, could be found. The other participants

Claim	'491 Claim Language	Gtalk
		could optionally move their characters to that address, where they would see a new graphical image, along with accompanying sounds, which is a multimedia message.
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	See Claim 3 above. Additionally, while using GAMECON and DOOM, Gtalk allowed users to send messages to each other which included both text and positioning information about characters, which information was translated by the DOOM game into visible occurrences on screen.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	See Claim 3 above. The web browser would operate in a separate subscreen. Also see claim 4 above.
6	The system of claim 4, wherein the other message is a multimedia message.	See Claim 3 above. The web page might contain both text and graphical images, and therefore would be a multimedia message. Also see claim 4 above. A message containing character positioning information which is translated to occurrences on screen, which might include images and sounds, is a multimedia message.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	"2.1.4 New User Login The User will be presented with Several Prompts which will collect their personal information." As is evident from the source code file useredit.c for the OS2 version of Gtalk, this software collects "Real Name", "Address", "City", "State", "Postal Code", "Birthdate", "Voice Phone", and "Data/Fax Phone". In the Unix version 1.6.4, the sysop could edit user profiles using the "/U" command, and then entering the hard-coded password "jomama!". Pressing 'N' then allowed the sysop to create a new user and to create a user of class "GUEST", then enter a "Name", "Street", "City", "State or Province", "Postal Code", ""Phone 1", "Pone 2", and "Birthdate".

Claim	'491 Claim Language	Gtalk
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	See claim 4.
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	Obvious in light of Donath, which discloses using world wide web browsers to encourage social interaction. See also Mosaic web browser and generally the World Wide Web.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the	See claim 1.

Claims	'491 Claim Language	Gtalk
	controller computer;	
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one	See claim 8.

Claim	'491 Claim Language	Gtalk
	member from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

82. If called to testify at trial I will be prepared to demonstrate the UNIX and DOS versions of Gtalk and testify about the source code of all versions, the Gtalk user manual, the Gtalk GAMECON feature, the DOOM video game software, and the testimony of Daniel Marks. I will also be prepared to make demonstrative exhibits from the above, such as by using screenshots, representations of screenshots, or other exhibits to demonstrate my opinions.

83. The paper titled "Sociable Web" by Judith S. Donath and Niel Robertson describes a system called "Sociable Web" whose main component is a system called "WebTalk". This paper appeared in the "Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web". The conference took place October 17-20, 1994, in Chicago, IL. Testimony in this case demonstrates that the paper was presented at the conference, and that this paper, as part of the proceedings, was made available on-line prior to the conference.

84. The description in the paper of WebTalk discloses a system that allows multiple visitors to the same web page to interact through a chat system. The chat system allows users to send both group and private messages, and the system allows users to send multimedia messages, including messages that include text, and links to web pages, images, audio files, etc. The paper provides images that depict windows shown on the screens of users of the system. The paper also teaches a variety of authentication mechanisms.

85. The charts below indicate how the asserted claims of the patent are disclosed in the paper by Donath and Robertson.

Claim	'491 Claim Language	The Sociable Web
1	Computerized human communication arbitrating and distributing system, including:	Donath is directed towards such a system (see below)
1(a)	a controller computer;	The computer serving as a WebTalk server is a controller computer. "A WebTalk server is a normal httpd server with some added capabilities: it keeps track of all the users on the pages it serves and it relays the data in the public

Claim	'491 Claim Language	The Sociable Web
		conferences to the participants."
1(b)	a plurality of participator computers	**Participator computer:** A computer running the "WebTalk client" or "Sociable Web browser" is a participator computer. **A plurality of participator computers:** Multiple "WebTalk clients" or "browsers" running on different participator computers may connect to a WebTalk server simultaneously. The paper uses the terms "user," "participator," and "person" to refer to a person that has connected to a WebTalk server via a WebTalk client running on a participator computer. These terms are also used in the plural ("users" and "participators"), indicating a plurality of participator computers. Furthermore, a message from a WebTalk client to a WebTalk server indicates which participator computer (called a host) is making the connection. "The Sociable Web project consists of a modified Web browser and server. The browser looks like an ordinary browser, and on pages not served by a Sociable Web server, it functions normally. On Sociable Web pages, however, it provides a number of social and collaborative features. Most notably, **it shows who else is on the pages** and it allows the user to strike up conversations or to join in ongoing discussions." "A WebTalk server is a normal httpd server with some added capabilities: **it keeps track of all the users on the pages** it serves and it relays the data in the public conferences to the participants." "A WebTalk client sends a message to the server whenever it arrives at or leaves a page. A non-WebTalk server ignores these messages; a WebTalk server acknowledges them, **letting the client know that it can look for other users on the page**. The WebTalk server uses these messages to keep track **of who is currently on its pages**. The message provides the user name, **host**, and WebTalk port number – all the information needed to establish contact with the person." "For public conferences, the server acts as a conduit; the user sends the message to the server, **which relays it to the other participants**. Messages are received with data identifying the sender and the discussion it was sent to (since one may be involved in several discussions at once)."
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	WebTalk users type messages using an input device such as a keyboard, and view messages on their video screens. "WebTalk discussions are live: one types a message and it appears instantly (or at least reasonably fast) on the screens of the intended recipients."

Claim	'491 Claim Language	The Sociable Web
1(d)	each said user having a user identity;	A WebTalk "callsign" is a user identity. "Since the connection is specified by machine name and port, one can use any name as a 'callsign'. It will be up to the server to determine whether visitor identity is authenticated and by what mechanism: this is part of establishing the general style of the server's conferences." "Other servers might wish to be more restrictive, permitting only the page owner or a chosen group of people to form (and dissolve) conferences and requiring that participants use their real (or at least, traceable) names."
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	WebTalk uses the TCP Internet protocol to connect WebTalk clients to WebTalk servers through the Internet. "The WebTalk port is a tcp socket that is kept open for data transfer: it is through this socket that the WebTalk discussions take place." "The message provides the user name, host and WebTalk port number, all the information that is needed to establish contact with that person." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants. Messages are received with data identifying the sender and the discussion it was sent to (since one may be involved in several discussions at once)."
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The WebTalk server software is the controller software. The paper also sometimes uses the term "Sociable Web server" of which the "WebTalk server is one example." "The Sociable Web consists of a modified Web browser and server." "A WebTalk server is a normal httpd server with some added capabilities: it keeps track of all the users located on pages it serves and it relays the data in the public conferences to the participants."
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group**: A "public conference" associated with a web page in WebTalk is a group. The paper also sometimes uses the term "discussion", of which a public conference is one example. "We are currently developing an experimental server and client that allows Web users to see who else is on a page, communicate with them, and travel around the Web as a group." "The Sociable Web system is based on the concept of shared location: you are able to talk only with other people who are on the same page." "The Sociable Web project consists of a modified Web browser and server. The browser looks like an ordinary browser, and on pages not served by a Sociable Web

Claim	'491 Claim Language	The Sociable Web
		server, it functions normally. On Sociable Web pages, however, it provides a number of social and collaborative features. Most notably, it shows who else is on the pages and it allows the user to strike up conversations or to join in ongoing discussions." "The main feature of the Sociable Web is WebTalk: the discussions that occur in the context of the Web and that use its rich hypermedia capabilities. For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants. Web talk discussions are live: one types in a message and it appears instantly (or at least reasonable fast) on the screens of the intended recipients. The **discussions can be public conferences**, open to all, or they can be private conversations between two people." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants" "A Sociable Web server should be able to determine the nature of the conferences that occur on its grounds." **Group through the controller computer:** Participants of a group (public conference associated with a web page) connect to the same controller computer (the WebTalk server that hosts the page). "The Sociable Web system is based on the concept of shared location: you are able to talk only with other people who are on the same page." "A WebTalk server is a normal httpd server with some added capabilities: it keeps track of all the users on the pages it serves and it relays the data in the public conferences to the participants." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants" **Plurality of groups through the controller computer:** A controller computer (WebTalk server) may host multiple web pages, each with its own group (public conference associated with a web page). "The Sociable Web system is based on the concept of shared location: you are able to talk only with other people who are on the same page." "A WebTalk server is a normal httpd server with some added capabilities: it keeps track of all the users on the pages it serves and it relays the data in the public conferences to the participants." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the

Claim #	'491 Claim Language	The Sociable Web
		other participants" Donath teaches that "one may be involved in several discussions at once." **Arbitration:** The paper teaches that the controller computer (a Sociable Web server) may arbitrate access to its groups (public conferences associated with the web pages that it hosts). "As for future work, there are several directions we see this work taking. One is developing the range of server styles. A Sociable Web server should be able to determine the nature of the conferences that occur on its grounds. Some might be very casual, allowing anyone to create a conference and permitting people to use any name as their identifier (this is for now the normal setup). Other servers might wish to be more restrictive, permitting only the page owner or a chosen group of people to form (and dissolve) conferences and requiring that participants use their real (or at least, traceable) names. These and other variations in server style will help a page owner create a social atmosphere that best matches the environment of the page." **In accordance with predefined rules including a test for an authenticated user identity:** "It will be up to the server to determine whether visitor identity is authenticated and by what mechanism: this is part of establishing the general style of the server's conferences." "Other servers might wish to be more restrictive, permitting only the page owner or a chosen group of people to form (and dissolve) conferences and requiring that participants use their real (or at least, traceable) names. These and other variations in server style will help a page owner create a social atmosphere that best matches the environment of the page."
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing**: "A WebTalk server . . . keeps track of all the users located on the pages it serves and it relays the data in the public conferences to the participants." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants." **Real time:** "WebTalk discussions are live: one types a messages and it appears instantly (or at least reasonably fast) on the screens of the intended recipients."

Claim	'491 Claim Language	The Sociable Web
1(i)	at least some of the user messages are multimedia messages.	A "phrase" is part of a typed message. A phrase may have an object (e.g., a picture or web link) attaentched to it by its sender. The recipient can view these objects by clicking on the phrase. A message may consist of multiple phrases, and hence may have multiple objects, potentially of different media types, attached to it. "Furthermore, the popularity of various live conferencing systems (e.g. IRC, the chatrooms of AOL, social MUDS, etc.) attests to the usefulness of real-time talk interfaces. Adding communicative abilities to Mosaic's easy access to many different types of media makes it possible to create conference sessions in which the users can insert hypertext links, sounds and images amidst their normal conversational text." "The Sociable Web allows people to see who else is on a page and to communicate with them (and to communicate not only with words, but with sounds, pictures, and links to other places.) "**WebTalk.** The main feature of the Sociable Web is WebTalk: the discussions that occur in the context of the Web and that use its rich hypermedia capabilities." "Images, sounds, and links to other pages call all be integrated with the flow of words. The WebTalk client includes several tools for fluency in hypertext conversation. For instance, the user can highlight a phrase and then, simply by clicking on a picture (or link) on any Web page, attach the chosen object to the phrase. When the phrase is sent, the recipient sees it as highlighted text; if the recipient clicks on it, he or she will receive the picture (or follow the link). "A WebTalk conversation can transcend smiley-faces. One can have an entirely library of eloquent pictorial – or auditory – interjections. And a WebTalk conversation can be completely interwoven with the vast resources of the Web." See also the figure in the paper with caption "Discussion window (try the buttons and links)." Images and text are shown in a message, as are hyperlinks.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The "WebTalk client" software (a modified browser) is the participator software operating on a participator computer. As in 1(b), there may be a plurality of participator computers. The WebTalk client software sends messages to a WebTalk server, which may arbitrate (as in 1(g)) distribute (as in 1(f)) these messages to other participator computers. "The Sociable Web project consists of a modified Web browser and server. The browser looks like an ordinary browser, and on pages not served by a Sociable Web Server, it functions normally. On Sociable Web pages, however, it provides a number of social and collaborative

Claim	'491 Claim Language	The Sociable Web
		functions. "A WebTalk server . . . keeps track of all the users located on the pages it serves and it relays the data in the public conferences to the participants." "A WebTalk client sends a message to the server whenever it arrives at or leaves a page. A non-WebTalk server ignores these messages; a WebTalk server acknowledges them, letting the client know that it can look for other users on the page. The WebTalk server uses these messages to keep track of who is currently on its pages. The message provides the user name, host, and WebTalk port number – all the information needed to establish contact with the person." "For public conferences, the server acts as a conduit; the user sends the message to the server, which relays it to the other participants."
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	"Images, sounds, *and links to other pages* can all be integrated with the flow of words. The WebTalk client includes several tools for fluency in hypertext conversation. For instance, the user can highlight a phrase and then, simply by clicking on a picture (or link) on any Web page, attach the chosen object to the phrase. When the phrase is sent, the recipient sees it as highlighted text; if the recipient clicks on it, he or she will receive the picture (or follow the link)." (emphasis added). See also Discussion Window in Sociable Web article.
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	Under Windy City's claim construction, a participator computer is compelled to locate another message when a user clicks on a link. Hence for Windy City's construction, see 3.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	A "window" in WebTalk is a "subscreen". A "discussion" is shown in a discussion window. The other message is displayed in a browser window. This window is distinct from the "discussion" window. See the figure with the caption "Discussion window (try the buttons and links)." "The Sociable Web project consists of a modified Web browser and server. The browser looks like an ordinary browser, and on pages not served by a Sociable Web server, it functions normally. On Sociable Web pages, however, it provides a number of social and collaborative features. "When the phrase is sent, the recipient sees it as highlighted text; if the recipient clicks on it, he or she will receive the picture (or follow the link)."

Claim	'491 Claim Language	The Sociable Web
6	The system of claim 4, wherein the other message is a multimedia message.	See claim 3 for definition of a "the other message" and "multimedia message" in this context. Since a link may lead to any web page, and a web page may be considered a "multimedia message", the other message may be a multimedia message. "Images, sounds, and links to other pages call all be integrated with the flow of words. The WebTalk client includes several tools for fluency in hypertext conversation. For instance, the user can highlight a phrase and then, simply by clicking on a picture (or link) on any Web page, attach the chosen object to the phrase. When the phrase is sent, the recipient sees it as highlighted text; if the recipient clicks on it, he or she will receive the picture (or follow the link)."
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	The paper suggests requiring participants to use their real names. A very straightforward implementation would be to store the real names, along with passwords, on the controller computer. Storing additional information about a participant at the controller computer is an obvious extension. "It will be up to the server to determine whether visitor identity is authenticated and by what mechanism: this is part of establishing the general style of the server's conferences." "Other servers might wish to be more restrictive, permitting only the page owner or a chosen group of people to form (and dissolve) conferences and requiring that participants use their real (or at least, traceable) names. These and other variations in server style will help a page owner create a social atmosphere that best matches the environment of the page."
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	The participator software (the WebTalk client program, see claim 2) is a modification of the Mosaic browser. The Mosaic browser automatically invokes different helper programs to view certain types of objects or when a link specifies certain protocols. For example, versions of Mosaic prior to 2.5 invoke the xview program to display JPEG images. Similarly, Mosaic invokes the telnet program when the protocol in the link is telnet (rather than http). Hence when a user clicks on a link, the browser may locate and then invoke a helper program such as xview or telnet. "Adding communicative abilities to Mosaic's easy access to many different types of media makes it possible to create conference sessions in which the users can insert hypertext links, sounds and images amidst their normal conversational text." "The Sociable Web project consists of a modified browser and server."

Claim	'491 Claim Language	The Sociable Web
		"When the phrase is sent, if the recipient clicks on it, he or she will receive the picture (or follow the link).
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	The participator software (the WebTalk client program) is an Internet browser (as it is a modification of Mosaic). The browser is invoked when a user clicks on a link. See Claim 26.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.

Claim	'491 Claim Language	The Sociable Web
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one member from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller	See claim 8.

Claim	'491 Claim Language	The Sociable Web
	computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

86. If called to testify at trial, I would be prepared to testify concerning The Sociable Web article and web browsers and supporting software it describes, including WebTalk, Netscape Navigator, and Mosaic and to present related exhibits. I will also be prepared to make demonstrative exhibits from the above to demonstrate my opinions.

87. WebTalk is a computer program written by Niel Robertson. It is described in part in the paper titled "The Sociable Web" by Judith S. Donath and Niel Robertson. The analysis in the preceding section (The Sociable Web) relies entirely on that paper. Although "The Sociable Web" article itself anticipates and enables all of the asserted claims of the '491 patent, the WebTalk software, which is the actual implementation of the system described in the article, also contained all of the features in the asserted claims. It was completed by Robertson in 1994, and was publicly disclosed at a Harvard conference by the end of 1994.

88. This section analyzes the WebTalk program, and is based on the deposition given by Niel Robertson on May 25, 2005. All comments marked with Q or A are taken from the deposition. Q: refers to a question asked by Mr. Hoover. A: refers to an answer by Niel Robertson. All comments in italics are my own.

Claim	'491 Claim Language	WebTalk
1	Computerized human communication arbitrating and distributing system, including:	*[WebTalk is directed at such a system.]*
1(a)	a controller computer;	*[The controller computer is the computer running the modified NCSA HTTPd web server]* Q: And what would you call the server side? Did you have a separate name? A: The server was HTTPD.
1(b)	a plurality of participator computers	**Participator computer:** *[A participator computer is the computer running the modified NCSA Mosaic web browser.]* A: The server was HTTPD. Q: Modified? A: A Modified version of it. The client was a modified version of Mosaic. **A plurality of participator computers:** *[A plurality of participator computers could run the modified browser, each making a connection to the controller computer Robertson uses the terms "client" and "modified browser" interchangeably.]* Q: Would that computer – let's talk about that particular architecture. That server really is talking, then to multiple

Claim	'491 Claim Language	WebTalk
		clients at the same time, is that correct, or roughly at the same time? A: Multiple clients are sending messages to that server, yes.
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	*[The program was programmed on and tested on a Sun Solaris workstation, which, in its standard configuration, comes with a keyboard, mouse, and video screen.]* Q: What operating system did this run on? A: I believe that I wrote the software on a Sun Solaris machine. Q: And that was for the client? A: Both. Q: Both the client and the server? A: Yes. Q: The -- and what was that the Unix operating System, a Sun version of Unix? A: Yeah, Solaris is Sun's version of Unix. Q: The computers that you actually used, at the time that you used them – this may be a silly question, but at the time that you used them to connect via TCP/IP to the server, did those computers have input or output devices, like a keyboard and monitor, for instance? A: Yeah. Q: They did? A: Yes, they did.
1(d)	each said user having a user identity;	*[As explained in 1(g), the modified HTTPd server could store a unique user name for each user along with an associated secret password.]*
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	*[The modified browser software running on the participator computer established TCP/IP connections with the modified HTTPd server running on the controller computer. TCP/IP connections are used to link computers on the internet.]* A: Sure. The way that a browser communicates with a server, a Web browser with a Web server, so Mosaic with the HTTP Daemon, is that when it wants something, it sets up a TCP/IP connection – and this is the state of the art at that time – Q: Sure. A: This evolved a bit. It sets up a TCP/IP connection initiated by the browser to the server, and then using the HTTP protocol, which has a very limited number of requests and response messages in it, it says basically, get me this thing... Q: Let's talk about the architecture with the server. That

Claim	'491 Claim Language	WebTalk
		was an architecture – TCP/IP, was that usable over the global Internet. A: Yes.
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	*[The controller software is the modified NCSA HTTPd web server.]* Q: So the finished product, WebTalk had perhaps two components, then, is that correct, a server side and a client side? A: That is correct. Q: And what would you call the server side? Did you have a separate name? A: The server was HTTPD. Q: Modified? A: A Modified version of it. The client was a modified version of Mosaic.
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	*[The controller computer maintained a list of who was visiting each web page hosted by the controller computer.* *Participants viewing a web page could create one or more (a plurality) of chat rooms associated with the page. Each of these chat rooms, together with the participants who joined the room, formed a group.* *The modified HTTPd server and Mosaic client supported user authentication through the use of usernames and password. This mechanism allowed arbitration as explained below.]* **Group:** *[Robertson calls a group a "conference room" or "chat room"]* Q: Can you tell me a little bit about the – some of the features of that software, what it did? A: It added the ability for a user using the client, using the browser, to see who else was looking at the same Web page they were looking at. It did that by extending the HTTP protocol with new messages, which essentially said this user as arrived and is looking at this page, this user has left this page. A combination of those two things, if you look at it, will allow you always to keep track of where somebody is. Additionally, when a user visited a page, the browser, using an extension to the HTTP protocol, would be able to download a list of everybody who was still on that page. So when the user was using the browser, they would see the traditional Web page. They would also see a list of all the people that that server still believed were on that

Claim	'491 Claim Language	WebTalk
		page...
		A: ... The second piece of technology that was added allowed anybody – there was no permission or security or anything in the version of the software I built. I just didn't get that far – allowed anybody to create a conference room, if you will, or a chat room associated with a web page.
		If I went to the espn.com homepage and I went to the main basketball area on ESPN's web site, I could set up a conference call, Colorado Nuggets fans, and anybody who then came to, was browsing through ESPN's site and was using my software, using the WebTalk software would see on that page all the conferences, including the Colorado Nuggets fans conference.
		You then could enter a conference and you could participate in a multimedia chat, discussion, whatever word you would like to use for it. I'll explain a little more about that in a second....
		Group through the controller computer:
		[As explained above, the controller computer kept track of the names of groups and which users were visiting which pages. It also kept a record of which groups had been formed on which pages. All of the participator computers connected to the controller computer.]
		A: When a user created a conference, that conference was associated with a specific Web page. So if you created a Conference A on be *[sic]* Web page 1, Conference B on Web page 2, all communication in and out of those conferences would be completely segregated. They would have no knowledge of each other.
		Q: This conferences page, would that be listing every conference that was on the server or would that be some subset of conferences that were on the server?
		A: The conferences page was relative to the currently viewed page in the Mosaic browser window in Exhibit 51.
		Q: Would that computer – let's talk about that particular architecture. That server really is talking, then to multiple clients at the same time, is that correct, or roughly at the same time?
		A: Multiple clients are sending messages to that server, yes.
		Plurality of groups through the controller computer:
		Q: Could a WebTalk server host multiple conferences?
		A: Yes.
		As you navigated to new pages, that list of conferences would reflect either zero, if there were none, or more conferences if there were, but only for the page that you're

Claim	'491 Claim Language	WebTalk
		reviewing. Q: So say the same server is hosting page 1 and page 2 and there are conferences going on each page. A: Yes. Q: Could there be multiple conferences based on the same underlying Web page? A: Yes. Q: How is that information conveyed to the user in the user's browser? Was it an architectural way, was it a push or was it a fetch command or how did it get there? A: When – as I mentioned, I extended the HTTP protocol. An HTTP protocol is based on a standard, so I extended the standard that was implemented with additional messages. One of those messages would have been the equivalent of tell me what conferences are on this page. What the software would have done is taken the current page and used that as part of the message to say, I'm looking at this current page. It would have sent a request saying, tell me all the conferences that are on the current page I'm looking at. There response to that message would have been zero or more conferences that were on that pages as far as the server understood it. **Arbitration:** *[Robertson indicates that his modified HTTPd server and Mosaic browser had all of the functionality of the unmodified server and browser. The server had long had support for user authentication, and a browser release months before he started the project also supported it.* *An entry page can be set up containing links to other, secretly named, pages, on which the chat groups were formed. Without downloading the links to those pages, unauthorized users would not be able to find the chat groups that they were not permitted to join.]* *[From the Mosaic User Authentication Tutorial, http://hoohoo.ncsa.uiuc.edu/docs/tutorial/user.html:* *"Mosaic 2.0 and NCSA HTTPd allow access restriction based on several criteria:* • *Username/password-level access authorization* • *Rejection or acceptance of connections based on Internet address of client* • *A combination of the above two methods* *...* *Before you can explore authentication, you need to install HTTPd 1.0a5 or later.* *So let's suppose you want to restrict files in a directory*

Claim #	'491 Claim Language	WebTalk
		called turkey to username pumpkin and password pie. Here's what to do: Create a file called .htaccess in directory turkey that looks like this: AuthUserFile /otherdir/.htpassword AuthGroupFile /dev/null AuthName ByPassword AuthType Basic <Limit GET> require user pumpkin </Limit> Note that the password file will be in another directory. ... Create the password file /otherdir/.htpasswd The easiest way to do this is to use the htpasswd program distributed with NCSA HTTPd. Do this: Htpasswd –c /otherdir/.htpasswd Type the password – pie – twice as directed. ... That's all. Now try to access a file in directory turkey – your browser should demand a username and password, and not give you access to the file if you don't enter pumpkin and pie. If you are using a browser that doesn't handle authentication, you will not be able to access the document at all."] [From NCSA Mosaic Version History, http://www.ncsa.uiuc.edu/Divisions/PublicAffairs/MosaicHistory/history.html "Version 2.0alpha3 Released April 6, 1994 ... • Access authentication"] [Although the tutorial indicates the HTTPd 1.0a5 is required, in fact, user authentication was already present in HTTPd 1.0a4, it had been replaced by the more reliable HTTPd 1.0a5.. HTTPd 1.0a2 introduced access authentication by IP address. From Upgrading NCSA HTTPd http://hoohoo.ncsa.uiuc.edu/docs/Upgrade.html: "HTTPd 1.0a5 • Fixed horrible bug in 1.0a4 HTTPd 1.0a4

Claim	'491 Claim Language	WebTalk
		• *Introduced user authentication (Basic scheme)* HTTPd 1.0a3 • .htaccess files now affect subdirectory *HTTPd 1.0a2* ... • *Introduced per-directory access by host and options control"]* *[NCSA HTTPd 1.0a5 was released 1993.* *"November 18, 1993* *New Form Creation/Submission Documentation to help people get started with the new forms capabilities of Mosaic 2.0, and NCSA httpd 1.0a5."* Q: Did the HTTPD server have the ability to a password protect a Web page? A: Well, the HTTP protocol at that point in time did have a limited facility for password protection. I could not tell you one way or the other if that version of the server had implemented it. My guess is yes, but I would be – Q: You're not sure. A: I'm not sure. Q: Is it correct to say that HTTP did have such a facility? A: Yes. Q: Can you describe for me the facility that it had for password protecting a web page? A: Sure. HTTP has a set of request messages and response messages. Response messages can either be positive with data or they can be a response for an error. One of the error conditions is a security condition is not met, such as a user does not have access to a directory. One of the request messages can include a user name and password that would give them access to that directory. Q: If you wanted to use a user name and password, how would it work at the time to protect a directory? A: I believe you – I believe there is a configuration file where you put a user name and password that were available for specific directories. If someone requested a file from that directory, the server would respond and say, I need a user name and password. The browser could ask for the user name and password and send it along with the request for the file to the server.

Claim	'491 Claim Language	WebTalk
		Q: The – can you describe for me, if you know, the concept of an HT access file. A: Yes. Q: Please tell me what that's about. A: An HT access file, I think, was an early user name and password file for directories that was implemented either for HTTPD or – that I used or versions around the time that I – from the one that I used to implement my extensions. Q: If that functionality was in the HTTPD server standard for Mosaic – from NCSA at the time, that functionality would have been part of your – A: Yes. Q: -- part of your modified program, correct? A: Yes. A: If you're asking me if I removed anything, then I'm saying no. And if you're asking me if it was in there, then, yes, it would still be in there. I did not remove anything from the HTTPD server.... **In accordance with predefined rules including a test for an authenticated user identity:** *[As explained above, a list of user identities along with the corresponding passwords can be stored on an NCSA HTTPd server to prevent users from accessing any web pages hosted by the server.]*
	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing:** *[All group messages were sent by the participator computers to the controller computer, which then distributed the messages pack to the participants who were the respective members of the group.]* A: there were two architectures – Q: Yes. A: – that the WebTalk software provided. One was for conferencing sessions, multiple people all talking in conference. The server – the HTML *[Robertson meant HTTP]* server would receive essentially a message that someone had added to the dialogue, and it would respond with the latest version of the dialogue. That way, multiple people could connect to the same server with their client and they could contribute to a conversation – it's very similar to discussion groups that we talked about before – and the server would give them whatever was relevant. A: Let me distinguish two situations. Situation one is where

Claim	'491 Claim Language	WebTalk
		a user was participating in a conference with multiple users. In that situation, the user would have maintained a TCP/IP connection to the server and that TCP/IP connection would have stayed running. Q: Persistent? A: Stayed persistent. That's a good word. And what would have happened is that on both the browser and the server, there would be what you call a listener, which listens on that connection for new information. And the server would be listening to the client to send it new content to add to the conversation, and the client would be listening for the server to send it updates or a new version of what the conversation looked like. Q: Listeners, they weren't human people? They were – A: No, no. A listener is a technical term for a piece of software code that listens on a computer connection for data to arrive. Q: And in this case of WebTalk, this would be – A: There would be a listener on the TCP/IP socket, is what it's called, but the TCP/IP connection, you'd constantly just ask the computer, did data arrive on this TCP/IP connection. It's a very common programming methodology. What would end up happening is that in a normal series of events, a user would enter text. The text would be sent to the server. The server would then recognize that data had appeared from one of the users in the conference. They would add it to the overall HTML, which was the conference conversation. They would then push that conversation – Q: You say "they." Would they – A: Sorry. The server would then push that conversation, the resulting conversation back through all the different clients that were – had persistent connection, and the client would then take that and present it in this first screen in Exhibit 50, in the conversation section of it. **Real time:** *[As in any chat room, the messages were delivered in real time.]* Q: So if we talk in the client-server mode, when the – when the message was received from one user, how long did it take before the message was sent back to the user that were participating in the conference? A: I believe that it was relatively real-time. And when I say relatively, whatever the time it took to process and then distribute that data back across the network. Q: A matter of milliseconds, something like this? A: It would be somewhere between milliseconds and a

Claim	'491 Claim Language	WebTalk
		second. Q: Sure. A: – two seconds. Q: Did you build in any type of delay to – such that the message would be delayed, the server side, so it would be from receipt of the message to distribution of the message. A: No. Q: There was no delay? A: There was no built-in delay. Q: And no built-in delay. So when the message was received from – let's say there was user one and user two and they were connecting in the client-server mode. Is it correct to say that each client is connected by its own TCP/IP connection to the server? A: That is correct. Q: And let's say user one decides to put a message. Just for simplicity, let's say it's a text message, sends a text message and wants that to be part of the conversation. When the user sends the – when user one sends the text message to the server, what does the server then do with it, with the text message? Let me ask you, does it distribute it back to both user one and user two? A: Yes. So the server would distribute the message to all the clients that were essentially connected with persistent connections, and the clients would add that or concatenate that to the overall conversation that they're maintaining. So you essentially would – if you sent the message, you would actually see it appear in your conversation via the server. Q: I see. A: So you would send a message off, and the server would say to all the clients, this was added to the conversation. The clients would present that to the user. ... A: So everybody, to my recollection, would see the exact same final conversation, regardless of any kind of network delays, or whatever the case may be, because the server would have said this message and this message and this message and this message as opposed to the client taking what the user entered and automatically putting it into the conversation irregardless of what the server told it. Q: And the server would do this as fast as it reasonably could; is that correct? A: Yes.

Claim	'491 Claim Language	WebTalk
		Q: Did the user – let's take user one, who sent the message. Did that user have to click any kind of refresh button in order to have the message be concatenated onto that use screen? A: I do not believe so. Q: So back to the exhibit, the discussion window image. I was going to ask you whether you saw any sort of refresh key in here. I think there isn't; is that right. A: There is no refresh key in the figure that I'm looking at. Q: Does the absence of a refresh key refresh you as to whether or no the user would have to hit refresh? A: I maintain my previous comment that it was an automatic feature of the software. Q: So – A: -- to refresh the conversation. Q: Let's say you have user one and user two, and we're still in the client-server mode. Let's say user one sends five messages to user two. Does user two get all five messages without having to do anything, basically? A: I would restructure what you said – Q: Okay. A: -- to be correct. User one would not send to user two in the client-server mode. User one would send to the conference, and the conference would distribute to all the other users, which would happen to be user two. You could have a conference with two users in it, right, but to distinguish client-server mode from client-to-client mode – Q: I appreciate that clarification. So we have – user one and user two are in a group in the client-server mode of operation. A: Yes. So when participant one in a conference sent the message to the conference, all the other participants would be pushed, without having to do anything else, the additional information that participant one sent.
1(i)	at least some of the user messages are multimedia messages.	*[WebTalk allowed participants to send anything that could be expressed in HTML to the group. Hence, anything that could appear on a web page, including an entire web page, could be sent to the group. The HTML could specify that the browser should render both text and graphical images together, by embedding the images on the page. The browser would fetch these images before rendering the page.]* A: In both of those cases, either going into a conference and joining a group of people talking about something or picking someone out of that list of users who were on a

Claim	'491 Claim Language	WebTalk
		Web page and having a direct communication, the software had essentially an interface in it that allowed you to do a few things. You could type in text and other HTML tags, such as links, images. Anything that HTML supported at that point in time, you could type that directly in. So you could – you could work with fonts. You could work with colors. Any of the presentation mechanisms of HTML, you could integrate in it if you wanted to. If you didn't, if you just wanted to type straight text, it would not require you to have that knowledge. Q: You could just type in straight text? A: You could just type in straight text and it would figure out how to wrap the appropriate HTML tags around it. The WebTalk would figure out how to integrate it into the final HTML that was presented to the other person or to the conference to make it work. You didn't have to have knowledge of HTML. The other piece of that interface was the resulting text or dialogue that you saw, and that would include, you know, who was making a discussion statement and all of the text, graphics, anything included in the HTML, and that would get sent to either the individual user or to the conference. In the conference session, you could – multiple people could be contributing at the same time, and the server would essentially orchestrate the conversation and organize who talked first, who talked second, who talked third, and then keep distributing that conversation out to all users. So you would type in information, and it would send it to the server, the server would add it essentially to the discussion, which was a long HTML page, would send it back to you, and you would present that HTML page aqnd it would look like a discussion thread, but it included all aspects of HTML, colors, fonts, layouts, graphics, etc. There was one other feature which was very – which was unique, which was all of the chatting input and output occurred in a separate window that was part of the overall application but appeared as a separate window. You could go back to the original browser and you could browse around and do whatever you wanted, and you would click on multimedia objects in that browser, a graphic a link, an HTML link, hyperlink to another page, and it would automatically insert that into your discussion, what you wanted to enter into the discussion. The reason for that is that most people don't know how to generate the HTML underlying tags to reference a graphic or reference a link and that if – the concept we had at the time was that you could take somebody on a tour.

Claim	'491 Claim Language	WebTalk
		You could say, go to this page, here, look at this graphic, and it would just put stuff into the discussion, and then what they saw coming back would be the graphics and links that you were clicked on. There's an integration, is what I'm saying, between the chatting windows and the browser, where you could directly pull more complicated multimedia elements out of the browser and have them be sent to your discussion, either directly to a person or to the conference, without having to know the underlying HTML programming language or coding language, if you will, you could still do more complicated things like graphics things. That was the gist of it. There were other simple things around, you know, opening conferences, closing conferences. Q: Very good. Sir, I appreciate that. When a first user, -- we talked a little bit – we talked about it a lot, actually, about how a first user would send a text message, again, in the client-server mode of operation to the conference. So user one – let's say there's users one, two and three in the next conference. It's pretty clear it's a client server mode of operation. So user one sends a text message, if your testimony is correct, the server sends that text message back to users one, two, and three, correct? A: Yes. Q: Let's say that – were there other types of messages that a user could send besides simply text? A: That the user could send to the conference? Q: To the conference. A: Well, a user – so let me be clear. The browser, the client would send a snippet of HTML. So to distinguish text from HTML, text as you think of it, is just plain text. You can represent something in HTML that looks like plain text to the user – Q: Sure. A: -- but behind it is HTML. Q: Understood. A: So the user – the client would always be sending snippets of HTML, which could have only text in it or it could have more complicated media, such as references to images or links, HTML hyperlinks. So the communication medium was HTML. The protocol was HTTP, and the messages inside the protocol were HTML snippets that would come from the client to the server and then be redistributed the way that Mr. Hoover described it.

Claim #	'491 Claim Language	WebTalk
		Q: And the ASCII text in some instances would be treated – well, let me strike that. Let's say, for instance, I wanted – user one wanted to send an image to the conference, just an image. How would the – how what that be accomplished. A: There are two ways that the user could manifest that. Let me use the discussion window figure as the example. Q: Sure. A: There is a graphic halfway down the conference session with the word "Web" a is part of the graphic. Q: That's under the name M.L. Saunders? A: Yes, they could either type into the entry-by-user portion, the interface that we talked about, an HTML reference to that image. I'll skip the description there of that, unless you want me to say it. They could also – using the feature I talked about in the earlier general overview, if they had navigated the browser window, and I'm referring to Exhibit 51, they had navigated the browser window to a page that had this graphic on it, they could click on that graphic, and all the HTML representation of how to find that graphic would be included for them into the entry-by-user portion of the interface. So they wouldn't have to understand the intricacies of HTML. They would then send – clicking the send button or something equivalent, they would send that. That message would go to the server, but to be clear it would be a reference to the image, not the image itself. The server would then redistribute that HTML snippet that got sent to it back to whatever participants were in the conversation, including the one that sent it, and the conversation part of the interface, which I'm pointing to on Exhibit 51, was an HTML rendering engine. It would know how to take the reference that was an ASCII to that image, go and get it and then display it, and what you would see here is the final figure. So all the messages back and forth between the server were in plain text. It was the magic of the HTML rendering engine, which I did not build, that would show the final graphic to the user. Q: Sure. Okay. Let's say I'm using M.L. Saunders. Let's just simplify it even further. You have the same three-party conference. Again, we're running the WebTalk software in the client-server mode of operation and a user wishes to send both an image and an ASCII text in the same message. Is that possible to do using WebTalk software? A: Yes.

Claim	'491 Claim Language	WebTalk
		Q: And could you describe to for me, please, in the same way you did before, how a user would go about doing that? A: Similarly to how I described putting a reference to an image into the conference, because the communication is with HTML, HTML can accept text, images, HTML hyperlinks, et cetera, interspersed, and it makes no distinction. So a user could have entered the HTML reference to that image, the Web image, in this figure and then types a question mark. They could have used the feature I described where they picked the image off a Web page and then typed a question mark, those two options. And then the final message that got sent would be the HTML combination of those things. So because it's HTML, you can mix and match however you like.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	*[The modified Mosaic browser running on the participator computer is the participator software.]* Q: So the finished product, WebTalk had perhaps two components, then, is that correct, a server side and a client side? A: That is correct. Q: And what would you call the server side? Did you have a separate name? A: The server was HTTPD. Q: Modified? A: A Modified version of it. The client was a modified version of Mosaic.
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	*[Embedded links that appeared in HTML messages sent to the group could be clicked on to optionally locate another multimedia message.]* Q: Sure. A link that was sendable in one of these groups or conference rooms, was – could that have been a link to another Web page on the Internet? A: Yes. Q: Could that other Web page have had text and graphics on it? A: Yes.
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	*[Under the plaintiffs claim construction, if a participant sends a URL message to the group, and the receipt of this message compels the participator computers of the members of the group to locate and present the corresponding web page, the terms of the claim are met. In WebTalk, as discussed in 1(i), a participant can send an HTML snippet containing a URL pointing to a graphical image embedded in an <IMG> tag, which compels the modified browser to locate and present the image, which is*

Claim	'491 Claim Language	WebTalk
		an other message.]
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	*[All user messages are displayed in a subscreen at the output device. This subscreen is the "chat and output" window that Robertson describes. Furthermore, within the chat and output window, the other message is displayed only in the output portion.]* Q: For Defendant's Exhibit 51, can you show please, the full screen with the various windows on it that a user might see,. A: Sure. You would have a – your traditional Mosaic browser, which if you installed Mosaic and you ran it, you would see this with all the traditional browser functions at the same time. A: You would have another window, which would be a window that I just drew up, which would be the two-pane chat input and output window. A: Yes. And you would have another window, I believe, which shows you virtual users, which would be all the users that are on the page that you're looking at. A: Sure. And you either had a separate or as part of the users on the page window – I don't remember which one it was – list of conferences available, sort of chatting conferences available on the page you're looking at.
6	The system of claim 4, wherein the other message is a multimedia message.	See claims 3 and 4.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	*[As explained in 1(g), HTTPd servers stored user names and their associated passwords. It would be straightforward to store additional information along with the user names.]*
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia	*[NCSA Mosaic, and hence the modified Mosaic browser had the ability to invoke different viewer programs to display media that could not be rendered by the browser itself. As an example, the Mosaic browser could not render Mpeg3 video, but if the participant clicked on a link to an Mpeg3 video that had been sent to the group by another participant, Mosaic would automatically invoke an Mpeg3 viewer and it would play the video.]* Q: Besides – well, we talked about the ability to send text and then graphics and links in a group, correct? A: Yes. Q: The links – was there some limitation on what the links

Claim	'491 Claim Language	WebTalk
	message at the respective output device.	could be links to? Could they be links – were they limited to the same server or limited to something? A: No. Let me define limitation. Web browsers have the ability to launch external programs to handle file types they don't inherently handle. An inherent file type that a browser would handle, for example, would be an HTML file. Back in 1994, an inherent file type that a browser would not have been able to handle would have been a movie, an MP3, those types of multimedia we're familiar with today. The user could configure the browser to launch an application on their system that would run the movie or play the sound, and if they clicked on that link, if they had configured the browser correctly, that external application would have essentially displayed the content of the link, where it's a physical display like a file or a move or a sound or anything in between. It would fall on the user to configure the browser properly, but there was no limitation to doing that, to my knowledge. Q: Let's say that you have users in a conference again in the client-server mode of operation. One user wants to send the other users a link to a movie of some type. At the time in 1994, were there movies that were available – or I guess, videos – I don't want connote like a Hollywood movie, but a video – A: Sure. Q: – that would put the video in a computer file format? A: Yes. Q: Can you name some that were around back then for movies? A: MPG, MPG3, MOV, WAV. Q: WAV was only for sound? A: I'm sorry, Wave files for sound. Q: Was AVI around back then? A: Yes, I believe so. Q: So certainly there's at least one file format that had sort of this movie capability, correct? A: Yes. Q: And some of these file formats that existed in 1994 play both video images and sound? A: I believe so. Q: So you'd have – let's say you wanted to have a video of somebody talking. You could have the person talking, and then the sound would be coordinated with the video image, correct? A: That's correct. Q: Let's say that in one of these conferences that you have on the WebTalk, user one wants to send a link to this movie to some of the other – to the other users in the conference. That's the hypothetical here. As as a factual matter, could your modified Mosaic browser natively render such a movie? A: Could mine or could a Mosaic browser? Q: The one that you modified. A: If you mean – when you say natively render, if you mean

Claim	'491 Claim Language	WebTalk
		embedded in the page of the conference show a movie, I do not believe the technology at that point could do that. Q: But was the user able – were the users able to see the movie if a link to such a movie was sent in the WebTalk browser? A: If the user clicked on that link and the browser was configured correctly, then the movie was available, absolutely. Q: Absolutely, right? A: Yes. Q: Was it a known thing how to configure a browser to launch and invoke an external application? A: Mosaic came preconfigured for common formats, such as movies and text files and telnet. Q: When you say preconfigured, can you elaborate a little bit, please? A: It would have just – the way that it works is it takes the three- or four-digit extension of the file, so .htm, .html, .mov, .avi, whatever you'd like, and it basically says, if the URL points you to this type of an item, here's the application on my computer that I'm going to pass it to and here's the way to pass it to it. So, for example, call an MPeg player if you see a .mpg file, and here's the way that I call it in the operating system, basically arguments to the executable for the MPeg movie player. A: Yes, I mean, the general – and this is a generalization. The general way that it would work is the browser would download the movie to the local computer. It would stick it on file system in a temporary working directory that the browser would use for these types of things. It would then invoke or instantiate the MPeg movie player application. And as part of that instantiation, it would tell them where the file was in the local machine that it wanted the movie player to run. Q: And then assuming the users in the conference had speakers, would the users then see and hear the content of the movie? A: Yes. Q: You said instantiate or invoke. Were you intending to use those as synonyms? A: Yes.
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	*[The participator software in WebTalk was a modified browser (Mosaic) that preserved all of the original features of Mosaic. Hence, the browser is invoked to render multimedia messages.* *Furthermore, as discussed in claim 26, another browser, which might be cable of rendering a multimedia data type, could be registered as the viewer for that type. The modified Mosaic browser would then automatically invoke the other browser to render this data type if the user clicked*

Claim	'491 Claim Language	WebTalk
		on a link to an object of that type.] A: The server was HTTPD. Q: Modified? A: A Modified version of it. The client was a modified version of Mosaic. Q: Was it a known thing how to configure a browser to launch and invoke an external application? A: Mosaic came preconfigured for common formats, such as movies and text files and telnet. Q: When you say preconfigured, can you elaborate a little bit, please? A: It would have just – the way that it works is it takes the three- or four-digit extension of the file, so .htm, .html, .mov, .avi, whatever you'd like, and it basically says, if the URL points you to this type of an item, here's the application on my computer that I'm going to pass it to and here's the way to pass it to it. So, for example, call an MPeg player if you see a .mpg file, and here's the way that I call it in the operating system, basically arguments to the executable for the MPeg movie player.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.

Claim	'491 Claim Language	WebTalk
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one member from the group	See claim 8.

Claim	'491 Claim Language	WebTalk
	consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

89. If called upon to testify at trial, I will be prepared to discuss the foregoing and to prepare and demonstrate exhibits based on the testimony of Mr. Robertson.

90. U.S. Patent 5,880,731, "USE OF AVATARS WITH AUTOMATIC GESTURING AND BOUNDED INTERACTION IN ON-LINE CHAT-SESSION", by Christopher A. Liles and Manuel Vellon, filed December 14, 1995, and issued March 9, 1999, invalidates several of the claims in the '491 patent. The abstract of the patent gives a good summary of the invention (my highlights):

> Avatars representing participants in a *graphic chat session* are periodically animated to produce a gesture that conveys an emotion, action, or personality trait. Each participant in the chat session is enabled to select one of a plurality of different avatars to represent the participant in a graphic chat session. Associated with each avatar is a bitmap file that includes a plurality of frames illustrating the avatar in different poses, actions, and emotional states. Selected frames are displayed in rapid sequence in accord with a script file to create an animation effecting each gesture. The same script file is used to define a gesture for all of the avatars in the chat session. *A selected gesture can be transmitted with a text message to convey the user's emotional state.* A gesture associated with the avatar is automatically displayed from time to time when the avatar is not otherwise gesturing or moving. *The user can determine participants in the chat session with whom the user will interact, e.g., by defining a proximity radius around the user's avatar or by selecting the specific participants from a list. Avatars of participants that are outside the proximity radius (or otherwise not selected) and messages received from them are not displayed on the user's monitor.*

91. The following charts show that U.S. Patent 5,880,731 (the '731 patent) invalidates several of the claims in the '491 patent at issue in this case.

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles at al.
1	Computerized human communication arbitrating and distributing system, including:	The '731 patent describes such a system.
1(a)	a controller computer;	The controller computer is the chat server that manages the chat session. "The modem also connects to a telephone line to convey

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		signals bi-directionally between computer 30 and a server at a remote on-line service to which other participants in a chat session are connected."
1(b)	a plurality of participator computers	**Participator computer:** The participator computer is the personal computer on which the software that allows the user to type messages and select avatars and gestures runs. "A monitor 38 is included for displaying graphics and text produced when an executable program is being run on the personal computer for use in connection with the present invention, for displaying a graphic chat session." In claims 13 through 16, this participator computer is called the "central processor". Claim 15: "The system of Claim 15, wherein the machine instructions executed by the central processor further enable the participant to selectively initiate an animation that conveys a desired motion and/or state of mind of the participant to another participant in the chat system." Claim 16: "The system of claim 15, wherein the animation selected by the participant to convey the desired emotion and/or state of mind of the participant is simultaneously activated in combination with a textual message that is transmitted by the participant." **A plurality of participator computers:** A plurality of participants may participate in a chat session simultaneously, each using a distinct participator computer. Claim 6: "A method for enabling a plurality of different gestures to be implemented by a plurality of different avatars that represent participants in an on-line graphic chat session,..."
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	"A display is provided for displaying a graphic representation of a virtual space in which the on-line chat session is occurring." A monitor 38 is included for displaying graphics and text produced when an executable program is being run on the personal computer for use in connection with the present invention, for displaying a graphic chat session." "Input can be provided to personal computer 30 using either a mouse 40 for manipulating a cursor (not shown) on monitor 38, which is used for selecting menu items and graphic controls displayed on the monitor by pressing an appropriate selection button (not shown) on the mouse, or by input entered by the user on a keyboard."

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		From claims 12 and 17: "(b) a display for displaying a graphic representation of a virtual space in which the on-line chat session is occurring;"
1(d)	each said user having a user identity;	The patent assumes that participants will connection through a commercial network service provider. At the time of the invention, such providers almost uniformly assigned each user an identity and allowed the user to select an associated password. "Use of the computer for communicating on-line with others has recently become much more popular with the increased awareness by the public of the Internet and of services provided by commercial service providers." "One of the more common options for enabling several users of an on-line service to interact is through a chat session." "When connected to an on-line service and participating in the chat session, the avatar selected by the user in character selection box 70 will appear in the virtual world or room with the avatars of the other participants. The virtual world is displayed in either a two-dimensional or three-dimensional mode. In addition, the user's identification or name will be added to the list of participants in the chat session." Claims 26 and 27: "(a) providing the participant with an identification of other persons participating in the on-line chat session;"
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	The patent indicates that the participants in a chat session are linked to a controller computer. "The modem also connects to a telephone line to convey signals bi-directionally between computer 30 and a server at a remote on-line service to which other participants in a chat session are connected." The '731 patent is not specific about which network is to be used (calling it merely a "network"), but points out the increasing awareness of the Internet. The patent indicates that the participator computer is of the type intended to run Windows 95™, which provides built-in support for connecting to commercial service providers using the TCP/IP protocols, e.g., through Windows Dial-Up Networking (DUN), which supports the PPP protocol over serial lines. "The present invention generally relates to the use of graphic representations of participants in a chat session, who are communicating using linked computers..." "Use of the computer for communicating on-line with others has recently become much more popular with the increased awareness by the public of the Internet and of services

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		provided by commercial service providers." "The system includes an interface to a network on which the on-line chat system is being run; the interface enables the participant to transmit and receive data over the network." "Although the personal computer is of the type intended to run Windows 95TM, it is contemplated that other types of personal computers, such as those made by the Apple Computer Corporation, will also be usable in executing software to implement the present invention." From claims 12 and 17, and 30: "(a) an interface to a network on which the on-line chat session is being run, said interface enabling the participant to transmit and receive data over the network;"
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	The patent indicates that the controller computer is a chat server. "The modem also connects to a telephone line to convey signals bi-directionally between computer 30 and a server at a remote on-line service to which other participants in a chat session are connected." "One of the more common options for enabling several users of an on-line service to interact is through a chat session." At the time the patent was issued, these chat systems (e.g., IRC, Gtalk, etc.) almost uniformly used controller software operating on a controller computer to manage chat sessions.
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller compute; and	**Group:** In describing existing types of chat sessions, the '731 patent uses the terms "session" and "room", which both refer to groups. "The virtual space in which each chat session occurs is sometimes referred to as a "room," since participants interactively communicate just as if they were meeting in a room. "Yet, it should be possible to selectively limit the group of participants with whom a person interacts so that only selected avatars in the chat session are seen by the person and so that only communications from the selected members of the group are observed by the person." **Group through the controller computer:** The patent indicates that the controller computer is a chat server. "The modem also connects to a telephone line to convey signals bi-directionally between computer 30 land a server at a remote on-line service to which other participants in a

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		chat session are connected." "One of the more common options for enabling several users of an on-line service to interact is through a chat session." **Plurality of groups through the controller computer:** "Depending on the subject matter of the chat session, a number of different, but appropriate avatars will be provided from which a participant may make a selection. For example, if participating in a chat session involving gardening, a participant might select an avatar that appears as a gardner..." "The present invention provides the participant with a number of predefined avatars that can be selected to represent the individual in a chat session for a particular subject." "Each chat session is normally monitored by a host." At the time the patent was issued, chat systems (e.g., IRC, Gtalk, etc.) typically supported a plurality of groups on a single controller computer. **Arbitration:** The patent discusses and proposes a number of arbitration methods. In discussing existing chat systems, the patent says: "In chat sessions involving a well-known personality, hundreds of people may join the session, but only the host and the moderator are active in the chat session, and all others are simply observers. However, provision may be made to enable questions previously submitted by the observers to be displayed to solicit a response from the guest. The host controls the chat session." "There are times when a participant in a chat session may wish to limit those with whom the person interacts. For example, if a discussion between two of the people involved in the chat session is of particular interest to a third party, the third person may not want to be distracted by communication transmitted from others in the chat session. In many cases, the participant may want to enable selected persons in the chat session to view his/her avatar and the messages that are sent to those persons; however this type of interactive control is currently not practical. Yet, it should be possible to selectively limit the group of participants with whom a person interacts so that only selected avatars in the chat session are seen by the person and so that only communications from the selected members of the group are observed by the person. Moreover, it would be preferable to select the members of the limited group that will be observed by the participant in a more graphical and natural manner. When two people want to speak privately in a room, they simply move away from the others in the

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		room so that their private conversation is not audible beyond the range of the other person with home they are conversing. A similar approach should be applicable to limit those with whom a person interacts in a graphic chat world. Currently, no conventional graphic chat session provides a technique to spatially select the avatars of others that the participant received. Providing this feature will enable a participant to perceive the avatars of those selected and to receive communications only from those members of the chat session that have been selected. The participant will not perceive the avatars or communications from those who are in the chat room, but were not selected." "Another feature of the present invention enables a user to selectively determine if distant participants in the chat session will be hidden from the user. If this menu item is selected, the user can thus limit the participants in a chat room session with whom the user will interact. In the preferred embodiment of the present invention, the host of the chat session determines the radius around each participant's avatar beyond which the avatars of other participants and the transmission from the other participants will not be evident to the user if the "hide distant members" (participants) menu option is selected by the user." "It is also contemplated that in subsequent preferred embodiments of the present invention, the user will be provided with further controls to limit the other participants and communications visible to the user. For example, the user can determine the participants with whom he/she will interact in a chat session by setting a proximity radius around his/her own avatar. Any avatars of other participants that are within the proximity radius will be "heard" and "seen" by the user. To determine the proximity radius, the user will select a menu item, causing a dialog box to be provided in which the user enters a nominal measure of the radius." "If the avatar is outside the proximity radius selected by the user, the logic proceeds to a decision block 218 to determine if the participant is in an exception list. In the current preferred embodiment, the exception list only includes the host for that chat session. However, it is contemplated that the exception list may also include the names (or other identification) of specific individuals with whom the user wants to interact in the current chat session." "Each chat session is normally monitored by a host. The host has control of the chat session and is provided with controls such as such in FIG. 14 in a dialog box 280. In this dialog box, the host can indicate that one or more selected members are to be treated as spectators or participants in the chat session, by choosing one of the radio buttons 282 or 284."

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		Claim 10: "The method of claim 6, further comprising the step of enabling a participant to perceive communications from another participant in the chat session only if the other participant is represented by an avatar that is disposed within a defined distance of the participant's avatar." Claim 20: "A method for enabling a participant in a graphic on-line chat session who is represented by an avatar to restrict communication with others participating in the on-line chat session, ..." (See also claims 21 through 27, which further elaborate on arbitration.) **In accordance with predefined rules including a test for an authenticated user identity:** The patent assumes that participants will connect through a commercial network service provider. At the time of the invention, such providers almost uniformly assigned each user an identity and allowed the user to select an associated password. "Use of the computer for communicating on-line with others has recently become much more popular with the increased awareness by the public of the Internet and of services provided by commercial service providers." "One of the more common options for enabling several users of an on-line service to interact is through a chat session." "When connected to an on-line service and participating in the chat session, the avatar selected by the user in character selection box 70 will appear in the virtual world or room with the avatars of the other participants. The virtual world is displayed in either a two-dimensional or three-dimensional mode. In addition, the user's identification or name will be added to the list of participants in the chat session." Claims 26 and 27: "(a) providing the participant with an identification of other persons participating in the on-line chat session;"
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Distributing:** The patent describes a "graphic" chat system. Chat systems typically distribute messages among their participants. All participator computers are connected to a chat server: "The modem also connects to a telephone line to convey signals bi-directionally between computer 30 land a server at a remote on-line service to which other participants in a

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
		chat session are connected." **Real time:** Chat systems deliver messages in real time.
1(i)	at least some of the user messages are multimedia messages.	"A selected gesture can be transmitted with a text message to convey the user's emotional state." A gesture is an animated image. "The frame numbers used in a predefined gesture are the same for all the avatars employed in a chat session for a particular virtual world or room. Typically, several of the frames are displayed rapidly in sequence on a participant's monitor to produce an animation conveying a specific gesture. As is well known to those skilled in the producing of cartoon animations, the rapid display of a sequence of frames in which a figure is portrayed in slightly different poses causes the figure to appear to move in an animated fashion." "Messages that are transmitted to the user are displayed and scrolled in the history pane. Text that has scrolled out of view in the history pane can be accessed by the user moving a scroll box 266 in a scroll bar 264 in the history pane." "The user can enter text to be transmitted to other participants in the chat session in the text box 150 as noted above." Both text and gestures can be sent together: "In the preferred embodiment, gestures are not embedded or associated with text messages that are transmitted by a participant for display to other participants. However, it is contemplated that a user will be enabled to select a gesture to accompany text that is transmitted for display to the other participants in the chat session. The gesture thus selected will provide emphasis of the user's emotional state in connection with the text message. Currently, in the preferred embodiment of the present invention, the user can select a gesture that indicates the user's emotional state in response to a prior communication within the chat session, for transmission without accompanying text, but a selected gesture and a text message can readily be transmitted together." Claim 5: "The method of claim 4, wherein the animation selected by the participant to convey the desired emotion and/or state of mind is displayed simultaneously with a textual message that is transmitted by the participant." Claim 16: "The system of claim 15, wherein the animation selected by the participant to convey the desired emotion and/or state

Claim	'491 Claim Language	U.S. Patent 5,880,731 Liles et al.
		of mind of the participant is simultaneously activated in combination with a textual message that is transmitted by the participant." A user can also type a URL and send it as a message.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	The participator software is the software that displays the graphics and text on the participator computer. It allows the participant to send and receive messages and gestures, and view them on the monitor. Messages sent to the controller computer enable it to distribute the messages to other users. "Although the personal computer is of the type intended to run Windows 95TM, it is contemplated that other types of personal computers, such as those made by the Apple Computer Corporation, will also be usable in executing software to implement the present invention." "A monitor 38 is included for displaying graphics and text produced when an executable program is being run on the personal computer for use in connection with the present invention, for displaying a graphic chat session." "The software that enables the participant to select an avatar and to participate in a graphic chat session can either be downloaded from the service, or might be distributed on a floppy disk or CD-ROM disk. After the software is downloaded or transferred from the floppy disk into personal computer 30, it can be executed by CPU 53, so that the user can make a selection of the avatar for use in a graphic chat session." Claim 12: "(d) a central processor for executing the machine instructions, said machine instructions, when executed by the central processor, causing the central processor to control the interface and the display so that; (i) an animation is provided for the avatar in the virtual space, said animation comprising a plurality of frames played in sequence so that the avatar appears to move within said virtual space..." "Messages that are transmitted to the user are displayed and scrolled in the history pane. Text that has scrolled out of view in the history pane can be accessed by the user moving a scroll box 266 in a scroll bar 264 in the history pane." "The user can enter text to be transmitted to other participants in the chat session in the text box 150 as noted above."

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	A user can type a URL and send it as a message. Upon receipt of such a URL, another user can then optionally copy the URL into the address bar of a browser and locate and display the corresponding web page. The web page might contain both text and images, making it another multimedia message. At col. 9, the patent teaches that the avatar is viewed as a static image until the other participant has downloaded the bitmapped avatar image. "Once the bitmap file for the user's avatar is customized, it can be selectively published, i.e., uploaded to the server maintained by the service on which the chat session runs, so that other participants in a chat session can download the customized bitmap file into hard drives of their computers. If a participant in a chat session has not downloaded the customized bitmap file of the user, when the user joins the chat session, the participant will see an amorphous ghost-like image that represents the user. Once the participant downloads the customized bitmap file for the avatar of the user, the user's customized avatar and gestures will be apparent to the participant."
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	Under the plaintiff's claim construction of "compel", which does not preclude user intervention, this limitation is satisfied by sending a message, such as a URL pointing to a web page, that the user can enter into a web browser, thereby compelling the message to be displayed in the web browser.
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	See claim 4. The subscreen is the browser window.
6	The system of claim 4, wherein the other message is a multimedia message.	See claim 4 and 5.
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	The patent already discusses storing a user's name and characteristic gesture. Storing additional information is an obvious feature to add.

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	To locate and invoke a separate computer program, such as a web browser, to process a URL that might be included in a user message, would have been obvious to one of skill in the art, since Mosaic and other web browsers had long provided the functionality of locating and invoking "helper" programs to process different types of data.
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	See claim 26.
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the	See claim 1.

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
	messages distributed by the controller computer;	
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an address is carried out with the other message including a multimedia message.	See claim 6.
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controlier computer. the authenticated user	See claim 8.

Claim	'491 Claim Language	U.S. Patent 5,880,731, Liles et al.
	identity including respective representations of at least one member from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

92. The paper "Integrating Communication, Cooperation, and Awareness: The DIVA Virtual Office Environment," by Markus Sohlenkamp and Greg Chewlos, describes a CSCW (Computer Supported Collaborative Work) system called DIVA. The paper appeared in the proceedings of the 5th ACM Conference on Computer Supported Cooperative Work, published by ACM Press. The conference took place October 22-26, 1994. It is my understanding that the paper was presented at the conference, and that the printed proceedings were distributed at the conference.

93. The charts below indicate how the asserted claims of the patent are disclosed in the paper by Sohlenkamp and Chewlos.

Claim	'491 Claim Language	DIVA
1	Computerized human communication arbitrating and distributing system, including:	DIVA is directed towards such as system (see below)
1(a)	a controller computer;	All DIVA applications are executed on a single computer. "The actual tools for working in DIVA are multi-user applications built with our GINA application framework [4]. A wide variety of prototype multi-user applications have been implemented in GINA in order to demonstrate its generic nature. These applications include a text editor, spreadsheet, structured drawing tool, music editor, and a chess program. Facilities to support synchronous group editing which are provided by every multi-user GINA application include group awareness in the form of visual representation of others' actions, unlimited multi-user undo/redo, multiple coupling modes, embedded annotations, optimistic concurrency control, and conflict resolution." "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process."
1(b)	a plurality of participator computers,	A "workstation" is a participator computer. Each user of the DIVA system sits at a separate workstation. "Cici, working at another workstation, is in the process of adding a rectangle to the drawing." "For the video conferences, a miniature camera attached to the top of each workstation sends an analog video signal to

Claim	'491 Claim Language	DIVA
		a special video board which displays the image in an X window."
1(c)	each said participator computer connected to an input device for receiving input information from a user and to an output device for presenting user messages,	Each user sits at an individual "workstation", a term for a personal computer with a video screen, mouse, keyboard, microphone, speakers, and video camera. Video screen: "Small video windows are then automatically opened and placed at the top of the screen, one window for each occupant of the room." Using mouse to drag and click: "So, in order to converse with another person in the DIVA virtual office, users simply drag their icon into the DIVA room where the target person is working, using the virtual office window. " "In DIVA, audio is temporarily suspended by clicking on the privacy button (the icon on the right end of the tool bar in the room window)." Using keyboard to edit text: "A wide variety of prototype multi-user applications have been implemented in GINA in order to demonstrate its generic nature. These applications include a text editor, spreadsheet, ..." Audio and video input/output: "For the video conferences, a miniature camera attached to the top of each workstation sends an analog video signal to a special video board which displays the image in an X window." "... when a DIVA user enters a virtual room already occupied by one or more users, audio and video links are established between the newcomer and the other occupants." "Audio channels are opened at the same time, and people already in the room are informed of the arrival by an audio cue." "During a private conversation in DIVA, the sound of the conversation is transmitted to others at a very low volume, while the sounds from the others are received normally."
1(d)	each said user having a user identity;	"People represent the users of the DIVA system and are implemented as snapshots with a name beneath." "As illustrated in the example, a glance at the virtual office window provides a broad level of awareness of co-worker activities: Markus, Cici, and Mike are together in Markus's office; Claus and Andreas have met in the project room; Greg and Thomas are each alone, but available for contact; the people in the "Conference" room would like some privacy; and user Jim does not want to be disturbed..."

Claim	'491 Claim Language	DIVA
1(e)	connections through the Internet linking the controller computer with each of the participator computers; and	"DIVA transmits these signals over the same TCP/IP network used for computer communications" In more detail: TCP/IP connections are made between the controller computer and each participator computer so that the controller computer can display an application running on the controller computer in an X window on the screen of a participator computer. (See also reference 36 in the paper.) Supporting quotations: "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process." "A typical DIVA session is illustrated in Figure 1 (henceforth referred to as "the example"). The virtual office, shown from the point of view of user Markus, is displayed in two main windows. The first window (in the background) contains the virtual office itself and the second (in the mid-ground) shows the virtual room that the user is currently in." "The foreground window is a shared graphics editor, currently in use by the three people shown in the video windows at the top." "For the video conferences, a miniature camera attached to the top of each workstation sends an analog video signal to a special video board which displays the image in an X window." The document indicates explicitly that the top three windows in Figure 1 are X windows. The three other windows on the workstation screen, which have the same frames, are also X windows. TCP/IP is also used to carry audio signals to "AudioFile" servers which are then relayed to the workstations. The AudioFile servers may operate on the same controller computer hosting the GINA applications "Unlike most media spaces which use separate analog networks for sending audio and video [e.g., 13, 28,33], DIVA transmits these signals over the same TCP/IP network used for computer communications. Audio connections are provided using AudioFile audio servers [24] and special client applications. The servers support the mixing of multiple input channels, which permits DIVA to combine voices from other users with its own audio cues."
1(f)	Controller software operating on and directing the controller computer to carry out the steps of:	"While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process."

Claim	'491 Claim Language	DIVA
1(g)	arbitrating in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	**Group:** A "room" in DIVA is a group. "*Rooms* are containers for people, desks, and documents. They also control the audio/video communication status of users. Just as people located in the same real room are able to see and hear one another, so too can people in the same DIVA virtual room hear and see each other; when a DIVA user enters a virtual room already occupied by one or more users, audio and video links are established between the newcomer and the other occupants." **Group through the controller computer:** "All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process." **Plurality of groups through the controller computer:** "Rooms can be used as private offices, public meeting places, or special purpose places." "Rooms themselves are contained in the DIVA virtual office environment. Users may customize their virtual office by selecting their set of potential cooperation partners and placing the rooms as they like. A glance at the rooms contained in the virtual office shows users who is inside each open room." "As illustrated in the environment, a glance at the virtual office window provides a broad level of awareness of co-worker activities: Markus, Cici, and Mike are together in Markus' office; Claus and Andreas have met in the project room; Greg and Thomas are each alone but available for contact; the people in the "Conference" room would like some privacy; and user Jim does not wanted to be disturbed, as indicated by the lock on his DIVA office." **Arbitration in accordance with predefined rules including a test for an authenticated user identity:** "Rooms also serve to indicate availability and communication willingness: they can be in different states, providing different levels of access and visibility of their inhabitants." "Access control, in one form or another, is an essential part of any multi-user environment. DIVA provides availability states for rooms which give users control over both their availability and the awareness information about them which is conveyed to others. It also includes access lists to give users control over the use of rooms and documents." "**Access Lists. DIVA** implements rudimentary object access control in the form of *access lists* for rooms and documents. The room access list determines which users are allowed to enter the room when it is locked. Conceptually, users on the access list have a key to the

Claim	'491 Claim Language	DIVA
		room. Visual feedback is provided in the form of small key icons on the rooms which the user has access to (e.g., the "Coffee room" in the example). Only users on the access list for a room are able to change the room access status. Typically, the only person on the access list of a private office is the owner of that room." "The document access lists controls document appearance and accessibility. Users on the access list of a document will see the icon in its normal form in the virtual room window and have full access to it. "Room and document access lists are managed similarly. Anyone on an access list may add others to the list while initially the list only contains the person who created the corresponding room or document. Finally, anyone on the access list of an object may set a special flag granting universal access to the object. In this case, all users may access the object. While much more complex access control mechanisms are possible, this simple mechanism is all that is needed in our prototype." **Predefined rules:** "**Availability Status.** In a manner similar to the door states use in the Ontario Telepresence Project [8], DIVA allows users of rooms to signal and to limit their availability for contact. Rooms may be *open*, *locked*, or *shuttered*. Open rooms may be entered by anyone, and their occupants are visible in the virtual office window. This state signals high availability for contact and provides the highest level of awareness. Locked rooms can only be entered by those with a key to the room (see below) and the occupants of the room cannot be seen without entering the room. The locked state indicates very low availability for contact and provides the maximum degree of privacy, at the expense of awareness. The shutter state provides an intermediate state between these extremes. The occupants of a shuttered room can not been seen directly but can be seen by moving to the threshold of the room. This causes the blinds to lift momentarily, sends an audio cue to the room occupants, and allows them to see their DIVA room windows who is glancing in. These access states are indicated visually, as illustrated in the example; most of the rooms are open, but Jim's room is locked and both Mike's room and the "Conference" room are shuttered." **Authenticated user identity:** DIVA implements access control through access lists, which implies stored access information and authentication. A user identity is authenticated to other users through the use of real-time video images and audio feeds. "For the video conferences, a miniature camera attached to the top of each workstation sends an analog video signal to a special video board which displays the image in an X

Claim	'491 Claim Language	DIVA
		window. A video server process sends the contents of this video to the video servers on the workstations of the other members of the video conference. " "Control of conferences is based on a very simple model taken from the real world – people in the same room can see and hear another while others cannot. So, in order to converse with another person in the DIVA virtual office, users simply drag their icon into the DIVA room where the target person is working, using the virtual office window. Small video windows are then automatically opened and placed at the top of the screen, one window for each occupant of the room. Audio channels are opened at the same time..." In addition, users typically must enter a name and password before operating a workstation (participator computer) that runs an X windows X server.
1(h)	distributing, in accordance with the predefined rules, the user messages in real time to the respective ones of the participator computers; wherein:	**Message:** "Notes" or "stick on notes" in DIVA are user messages. Tables 1 and 2 show the correspondence. In Table 1, the cell in row "**Communication**" and column "**Asynchronous**" reads "Leave messages for others". In Table 2, the cell in row "**communication**" and column "**Asynchronous**" reads "leave notes for others wherever they are needed." "DIVA support for asynchronous communication is based on another object from the real-world office: the stick-on note. Notes can be attached to the objects in the virtual office: people, rooms, desks, and documents. To do so, a user drags the note tool onto the target object (the second icon on the tool bar of the room window in the example). This causes the note editor to pop up, which permits both creating new notes to attach to the object or reviewing existing notes on the object. After the new note is created, a note icon appears on the object, with one exception. Just as we do not actually stick notes on people in the real world, in DIVA notes directed at people do not appear on their icons but instead appear on their briefcases, where they are both private and accessible to the recipient. **Distributing:** Notes are stored on the controller computer. They are distributed to the participator computers by showing them in X windows displayed on the participator computers' screens. "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process." **Real time:**

Claim	'491 Claim Language	DIVA
		The paper categorizes notes as "asynchronous" communications, meaning that the entire note must be composed by one user before it can be viewed by any other user. Upon its completion, however the note appears in real-time. In particular, when a note is attached to a document, it changes the appearance of the document's icon. "After the new note is created, a note icon appears on the object..." Changes to documents, however, are displayed "synchronously" which is also equated with real-time. Table 1 (sans caption) is reproduced below: **Synchronous** / **Asynchronous** **Communication**: Communicate in real time. / Leave messages for others. **Cooperation**: Simultaneous work using groupware tools. / Turn-taking work. **Awareness**: What are others doing now? / What have others done recently? Note that the cell in the row labeled "**Communication**" and the column labeled "**Synchronous**" mentions "real-time". The corresponding cell in Table 2 indicates that this cell refers to audio and video communications. The cell labeled "**Cooperation**" in the column labeled "**Synchronous**" refers to documents, with the corresponding cell in Table 2 listing "manipulate (create, edit, etc.) shared artifacts." This cell indicates that any changes to a document are viewed synchronously, i.e., in real time. In particular, the attachment of a note to a document is viewed in real time. "**Synchronous.** The virtual office window provides a broad overview of co-workers' activities throughout the virtual office, while the virtual room window provides more detailed information about a particular room... Much of the information can be perceived even when not being actively attended to, such as the animated movement of people and documents..." "The document icons visually indicate the status of the document..." "Notes left on objects by others are shown as yellow squares on the corner of the objects. Notes are on the "Coffee room," the briefcase, and the "Song" and "Drawing" documents in the example. Visual cues are thus provided at both the office level and the room level, indicating what others have do0ne that is of interest to the user." (See

Claim	'491 Claim Language	DIVA
		Figure 1.) **Message:** A message can also take the form of a "document". A user can make a document viewable by other users by bringing it into a room and placing it on a desk where other uses are working. The document serves as a message to the other users in the room that have not yet viewed it. In this scenario, no further editing (shared or otherwise) of the document takes place in this scenario. "*Documents* represent the artifacts people work on in the virtual office." "Rooms are containers for people, desks, and documents...." "By moving shared documents and themselves to a desk in the room, users may work together in either tightly coupled or loosely coupled mode." "The other DIVA window, labeled "Room Markus" in the example, is the virtual room window. It reveals the contents of the room that the user is currently in. In addition to the people who are in the room, the desks and documents in it are shown." "To edit a document found in a virtual room, the DIVA user drags the corresponding icon to a desk." "DIVA users know from the visual clues on a document icon if a document has changed before they open it. On opening a changed document, users are given the opportunity to catch-up to the changes before proceeding with their editing." **Distributing:** Documents are stored on the controller computer. They are distributed to the participator computers by showing them in X windows displayed on the participator computers' screens. "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process." **Real time:** The discussion of the real-time delivery of notes also explains that changes to documents (including their arrival at a desk) are displayed "synchronously" or in real-time. "**Synchronous.** The virtual office window provides a broad overview of co-workers' activities throughout the virtual office, while the virtual room window provides more detailed information about a particular room... Much of the

Claim	'491 Claim Language	DIVA
		information can be perceived even when not being actively attended to, such as the animated movement of people and documents..." **Message:** A DIVA user can also send a real-time audio message to the other users in a room. "Real-time person to person communication communication is supported in DIVA through audio/video conferencing... Audio channels are opened at the same time, and people already in the room are informed of the arrival by an audio cue." **Distributing:** Audio messages are distributed through the controller computer. In particular, TCP/IP is used to carry audio signals to "AudioFile" servers which are then relayed to the workstations. The AudioFile servers may operate on the same controller computer hosting the GINA applications "Unlike most media spaces which use separate analog networks for sending audio and video [e.g., 13, 28,33], DIVA transmits these signals over the same TCP/IP network used for computer communications. Audio connections are provided using AudioFile audio servers [24] and special client applications. The servers support the mixing of multiple input channels, which permits DIVA to combine voices from other users with its own audio cues." **Real-time:** "Real-time person to person communication communication is supported in DIVA through audio/video conferencing... Control of conferences is based on a very simple model taken from the real world – people in the same room can see and hear another while others cannot. So, in order to converse with another person in the DIVA virtual office, users simply drag their icon into the DIVA room where the target person is working, using the virtual office window. Small video windows are then automatically opened and placed at the top of the screen, one window for each occupant of the room. Audio channels are opened at the same time..." The cell in the row labeled "**Communication**" and the column labeled "**Synchronous**" in Table 1 reads "Communicate in real-time", while the corresponding cell in Table 2 lists "make and break verbal and visual contact with one or more other people."

Claim	'491 Claim Language	DIVA
1(i)	at least some of the user messages are multimedia messages.	User messages may take the form of notes, documents, or audio messages. **Notes:** "Notes can be read by clicking on them or by using the note tool. The note editor and viewer supports text, audio, and video notes." **Documents:** A document might contain both graphical images and text. Figure 1 shows that three people are viewing a document in a shared graphics editor. "The foreground window is a shared graphics editor, currently in use by the three people shown in the video windows at the top." The graphics editor shown in Figure 1 has a button labeled "A". This label is a standard way of indicating that pressing the button will allow a user to add text to a document. If one user were to bring a multimedia document created using the graphics editor into a room and place it on a desk, the other users in the room could then view (and edit) it. **Audio**: A user can deliver a text-only note or document to a desk while at the same time speaking over the audio channel to the other users in the room. The combination of the text-only note or document and the audio communication forms a multimedia message.
2	The system of claim 1, further comprising: participator software respectively operating on and directing each of the participator computers to enable one of said users to send one of the user messages to the controller computer and to enable arbitrating and the distributing of the one of the user messages.	DIVA requires the X Windows "X server" software to operate on the participator computers. "For the video conferences, a miniature camera attached to the top of each workstation sends an analog video signal to a special video board which displays the image in an X window." The document indicates explicitly that the top three windows shown in Figure 1 are X windows. The three other windows on the workstation screen, which have the same frames, are also X windows.
3	The system of claim 1, wherein: the user messages include an address to instruct the participator computers to optionally locate another multimedia message.	A document or a note may contain a URL. Using the mouse in the X-window system, a user may highlight the URL and then paste it into the address bar of a browser to optionally locate another multimedia message. Also, a user deliver a document with a note attached to it a desk in a room. One of the two (e.g., the note) can be considered the first user message, whereas the document icon displayed by the X-windows software on the participator computer is an address to instruct the participator computers to optionally locate another multimedia message (by clicking on the icon to open the document).

Claim	'491 Claim Language	DIVA
4	The system of claim 1, wherein: the user messages include an address to compel the participator computers to locate an other message and to present the other message at the output device.	The paper anticipates taking advantage of GINA's decentralized architecture so that software could be invoked on the participator computer. "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process. In the future, the implementation will be changed to match the replicated model provided by the GINA framework."
5	The system of claim 4, wherein: the other message is displayed in a subscreen at the output device.	As Figure 1 shows, each X-window is a subscreen on the output device. A user message such as a document or a note is displayed in an editor in its own subscreen, such as the graphics editor or the notes editor. For example, in Figure 1 the document labeled "figure" is shown in its own window. "The foreground window is a shared graphics editor" "Markus, Cici, and Mike are all working on the shared drawing "figure" (shown in the graphics editor window) "To do so, a user drags the note tool onto the target object (the second icon on the tool bar of the room window in the example). This causes the note editor to pop up, which permits both creating new notes to attach to the object or reviewing existing notes on the object."
6	The system of claim 4, wherein the other message is a multimedia message.	A graphics document can be a multimedia message. "The foreground window is a shared graphics editor"
8	The system of claim 1, wherein: the authenticated user identity is stored at the controller computer, and the authenticated user identity includes at least two members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	At the time the paper was written (and to the present day), a common form of access control was through a stored user name and password, and a login procedure. Unix systems that ran the X windows X server software at the time that the paper was written stored authenticated user identities one-per-line in a password file, which typically included the user's login name (which served as the email address on the system) and the users' real name (in addition to a hash of the users' password). Administrators were also free to store other information about the user in the line and it would have been obvious to do so.

Claim	'491 Claim Language	DIVA
26	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computers; and invoking the computer program to present the multimedia message at the respective output device.	The paper notes that GINA applications (such as the graphics editor) are designed to run on the participator computer, rather than on the controller computer. To view a graphics document message, then, the graphics editor on the participator computer would be invoked. "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process. In the future, the implementation will be changed to match the replicated model provided by the GINA framework."
27	The system of claim 2, wherein: the participator software presents the multimedia message on the respective output device by steps including: invoking an Internet browser to obtain and present the multimedia message on the respective output device.	The paper notes that a variety of other applications have been implemented in GINA. As these other applications would be invoked on the participator computer in order to view multimedia messages such as graphics documents, it is obvious that a browser could also be invoked. "The actual tools for working in DIVA are multi-user applications built with our GINA application framework [4,37]. A wide variety of prototype multi-user applications have been implemented in GINA in order to demonstrate its generic nature. These applications include a text editor, spreadsheet, structured drawing tool, music editor, and a chess program." "While the multi-user GINA applications are based on a replicated architecture (users run their own copies of the application), DIVA itself is currently centralized. All data about the virtual world is contained in a single database, and all DIVA applications are started from the same Lisp process. In the future, the implementation will be changed to match the replicated model provided by the GINA framework."
40	A method for using a computer system to arbitrate and distribute human communication, the method including the steps of:	See claim 1.
40(a)	connecting a plurality of participator computers with a controller computer through the Internet,	See claim 1.
40(b)	each said participator computer for connecting to an input device to receive input information from a user and to an output device to present user messages,	See claim 1.
40(c)	each said user having a user identity;	See claim 1.

Claim	'491 Claim Language	DIVA
40(d)	programming the controller computer to control communication of the messages between the participator computers;	See claim 1.
40(e)	programming the participator computers to enable sending respective ones of the messages to the communicator computer and receiving those of the messages distributed by the controller computer;	See claim 1.
40(f)	arbitrating with the controller computer, in accordance with predefined rules including a test for an authenticated user identity, which ones of the participator computers can be a member in one of a plurality of groups through the controller computer; and	See claim 1.
40(g)	distributing with the controller computer, in accordance with the predefined rules, the messages in real time to the respective ones of the participator computers,	See claim 1.
40(h)	wherein at least some of the user messages are multimedia messages.	See claim 1.
42	The method of claim 40, wherein the step of distributing includes distributing an address to an other message.	See claim 1.
43	The method of claim 40, wherein the step of distributing includes distributing an address to another message and instructions requiring at least one of the participator computers to carry out the step of locating the other message at the address.	See claim 4.
44	The method of claim 43, further comprising the step of: displaying some of the other message in a subscreen at the output device.	See claim 5.
45	The method of claim 43, wherein the step of distributing an	See claim 6.

Claim	'491 Claim Language	DIVA
	address is carried out with the other message including a multimedia message.	
47	The method of claim 40, where in the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least one member from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
48	The method of claim 40, wherein the step of arbitrating is carried out by: storing the authenticated user identity at the controller computer, the authenticated user identity including respective representations of at least three members from the group consisting of age, telephone number, fax number, name, company, postal address, E-mail address, and URL.	See claim 8.
63	The method of claim 47, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: locating a computer program on a memory accessible to the respective one of the participator computer; and invoking the computer program to present the multimedia message at the respective output device.	See claim 26.

Claim	'491 Claim Language	DIVA
64	The method of claim 48, wherein the step of programming the respective participator computers includes programming the respective participator computers to present one of the messages as the multimedia message on the respective output device by steps including: invoking an Internet browser to present the multimedia message at the respective output device.	See claim 27.

94. If called upon to testify at trial, I would be prepared to discuss the DIVA paper and the aforementioned applications and supporting software.

INVALIDITY OF THE '491 PATENT FOR FAILURE TO DISCLOSE BEST MODE

95. The '491 patent indicates that a byte-code implementation is the preferred embodiment, and includes screen shots of a Java applet embodiment, but instead attaches the "telnet" embodiment to the patent, and not the Java embodiment.

96. I will assume that the code provided to me by WCI (Ex. 30) is the Java implementation that the screen shots were taken from, and that the file creation dates are correct. At least one of the features of the Java embodiment isn't described in patent. The Java embodiment uses a library of Java routines called "GIF factory" that were written by someone other than the inventor.

NON-CUMULATIVE NATURE OF GTALK

97. The inventor failed to bring Gtalk to the attention of the patent examiner. In my opinion, however, Gtalk is closer to the invention than other prior art cited in the patent. In particular, for example, there are great similarities between Gtalk and the "telnet embodiment" code attached to the patent. First, the inventor is a co-author of Gtalk. Second, the code attached to the patent includes code written by the inventor that was previously included in Gtalk. Files list.c and list.h are two examples. Third, the high-level structure of the two programs is the same. Both run a "server" process on the controller computer, and both run a "client" process for each user on the controller computer. Both use the same "token" structure for communication between the server process and the client process. Both provide a telnet interface. Fourth, Dr. Marks, in his deposition, could not come up with any explanation for why Gtalk did not invalidate Claim 35 of the patent. Also, the changes required to give Gtalk (e.g., Gtalk version 1.6.4 for Unix) what Marks calls the "multimedia" functionality present in the telnet embodiment, *i.e.*, the ability to send specially tagged URLs are minimal. None of these things is true for any of the cited prior art references.

98. The modifications to Gtalk are straightforward. All that is necessary is to add a new message type to Gtalk (e.g., type "URL"). The "client" component of the software would then send a message of type URL if the client terminated the line by pressing the control-u key rather than the "enter" key. Upon receipt of a message of type URL, the client component would modify the tag indicating the sender of the message from the normal tag such as "#02: (bruce)" to a URL tag such as "URL from #02:(bruce)". The "server" component of Gtalk would require even fewer changes. It would simply treat URL messages in the same way that it treats normal messages.

99. I modified Gtalk version 1.6.4 for UNIX so that users could send and receive URL messages. In order to properly receive URL messages, I inserted the following line into a list in Client/channelcli.c:

```
{ "URL", receive_URL, T_CH_MESSAGE},
```

and then had to change the 19 in the following line to 20:

```
token_list client_channel_tok = {19, client_ch_tokens };
```

I then created a copy of function "receive_message", in Client/channelcli.c, calling it "receive_URL", and made a single line change to it, replacing the line

```
sprintf(s,"#%02de:%c%s|*r1#%c %s | *r1",
```

with

```
sprintf(s,"URL from #%02de:%c%s|*r1#%c %s | *r1",
```

Note that I have merely added the characters "URL from". I made a similar change in Client/ddial.c, creating a new function ddial_receive_URL from ddial_receive_message, and again modifying a single line.

100. In order to allow a user to send a URL message by ending a line with Control-U rather than by pressing return, I made a few more changes. In Client/input.c, I added three lines to function get_input. First, I added a line

```
case 21:
```

right after the line

```
case 13:
```

Then just prior to the end of the "case 13" section, I added the lines:

```
if (nextchar ==21){dest[pos++]=21,dest[pos]=0);
```

Next, in Client/channelcli.c I made a copy of function "write_to_channel", calling the new function "write_url_to_channel", and defining it in channelcli.h. The only difference between the two is that I replaced the line:

```
"MESSAGE %s %s", channel, message);
```

with

```
"URL %s %s", channel, message);
```

Finally, in function main_loop in Client/gtmain.c, I copied the block of code that calls write_to_channel, and modified two lines. I changed

```
else if (*s) {
```

to

```
else if (*s && (s[strlen(s)-1] ==21)){ s[strlen(s)-1] = 0;
```

and I then called write_URL_to_channel rather than write_to_channel.

101. On the server side, I added the following line to Server/srv_channel.c:

```
{ "URL", distribute_URL, T_CH_MESSAGE},
```

and changed the 11 to 12 in the following line:

```
token_list server_channel_tok = { 11, server_ch_tokens };
```

I then made a copy of distribute_message, renamed it distribute_URL, defined this new function in Server/srv_channel.h, and modified a single line, changing

```
"MESSAGE %s %lu/%d %s",
```

to

```
"URL %s %lu/%d %s"
```

102. The program worked as expected. Lines terminated by control-u were recognized and processed by both the client and the server as a type of message ("URL") distinct from the normal message type ("MESSAGE"), and upon receipt were designated as such by the "URL from" string.

103. I spent about four hours making these changes and testing and debugging the program.

104. I declare under penalty of perjury that the foregoing is true and correct and reflects my opinions on the discussed subjects.

Bruce M. Maggs