SYSTEM AND METHOD FOR INTERACTIVE VIDEO CONFERENCING

Inventors: Eric C. Anderson, Gardnerville, NV (US); Michael Noriega, Sacramento, CA (US)

Appl. No.: 13/008,001
Filed: Jan. 17, 2011

Publication Classification
Int. Cl. H04N 7/14 (2006.01)

Video Conferencing System Diagram

World Wide Web 101

Database Server 109
Web Server 102

Video Streaming Server 103

110

Participant Internet Device 104
Participant Internet Device 105
Participant Internet Device 106
Participant Internet Device 107

Wired/Wireless Internet Connections 110

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participant 108

Participants (108) interact with the system through their Internet devices (104-107). The Video Conferencing System Diagram illustrates the connection between the World Wide Web (101), Database Server (109), and Video Streaming Server (103). The diagram shows how the system facilitates interactive video conferencing, where participants can connect through their Internet devices and interact with the system through the World Wide Web.
Figure 2 User Internet Device Software/Hardware Stack

- Conferencing Application 206b
- Virtual Machine Application 205b
- Operating System 203
- Hardware Drivers 202
- Hardware 201
- Downloaded Conferencing Application 206a
- Virtual Machine Browser Plug-in 205a
- Internet Browser 204
Figure 3 Web Server Software Stack

- Web Application 305
- Downloadable Conferencing Application 304
- Web Server 304
- Operating System 303
- Hardware Drivers 302
FIG. 8

Message Pending...

Spiritual Life Development Group Session: Forgiveness

Participants
- Grant or Peake, Floor
- Hilda Speakman
- John Beck
- Melinda Timmer
- Lisa Jones
- Sam Riley

Participants Status Icon

Click Button or Press Speaker for Action

Questions/Comments from Students

Text Input Field
Type your text questions or comments here

Press Text to Enter Exit Text Entry Mode
Send Message
Clear Message

My thoughts are we should continue to work through all of the questions one at a time...

Action Button

Text Entry Area

Status Panel

804
805
806
807
808
Figure 11: Action Commands

<table>
<thead>
<tr>
<th>Request</th>
<th>Facilitator VAT Preferred</th>
<th>Participant Data Stream Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VAT</td>
<td>AT</td>
</tr>
<tr>
<td>Conference Start</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Add Participant</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Raise Hand</td>
<td>X (5)</td>
<td>X</td>
</tr>
<tr>
<td>Lower Hand</td>
<td>X (5)</td>
<td>X</td>
</tr>
<tr>
<td>Give Floor (transfer)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Release Floor</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Take Floor</td>
<td>X</td>
<td>X (1)</td>
</tr>
<tr>
<td>Text Message</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Answer Text</td>
<td>X</td>
<td>X (6)</td>
</tr>
<tr>
<td>Extension</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Lock</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Drop (3)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Subgroup Done</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create Subgroups</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Breakout Subgroups</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Join Subgroup</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Exit Subgroup</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Reconvene</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Unlock</td>
<td>X (2)</td>
<td>X</td>
</tr>
<tr>
<td>Move Participant</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Clear Hands</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1) Only when no one has the floor
2) Only when subgroups not active
3) Generated by system when participant disconnects
4) Text only participant can only send text messages and not take floor
5) Optional - not included in preferred embodiment
6) Only when they have the floor
1200 1201
RECEIVE CONFERENCE START REQUEST & SCHEDULE FROM PARTICIPANT

1202 IS REQUEST VALID?
NO 1203 EXIT
YES

1204 RECEIVE DATA TYPE FOR PARTICIPANT (TEXT, TEXT+AUDIO, TEXT+VIDEO)
TEXT ONLY 1205 SL
VIDEO OR AUDIO

1205 SET CURRENT GROUP = GROUP OR SUBGROUP OF THE PARTICIPANT, RECEIVE DATA STREAM OF PARTICIPANT DATA TYPE FROM THE PARTICIPANT, SET PARTICIPANT AS SPEAKER OF CURRENT GROUP, SET ALL OTHER PARTICIPANTS IN CURRENT GROUP AS LISTENERS; SEND DATA STREAM TO LISTENERS IN CURRENT GROUP OR SUBGROUP; CLEAR ALL HANDS UP IN CURRENT GROUP OR SUBGROUP; DISPLAY PARTICIPANT IN CURRENT GROUP OR SUBGROUP PARTICIPANT LIST

1206 MAIN EVENT LOOP

1207 ADD PARTICIPANT REQUEST?
NO 1208 IS CONFERENCE OVER?
YES 1209 EXIT
NO

1208 IS REQUEST INVALID OR CONFERENCE LOCKED?
NO 1211 IGNORE
YES

1212 RECEIVE DATA TYPE FOR PARTICIPANT (TEXT, TEXT+AUDIO, TEXT+VIDEO)

1213 PARTICIPANT TYPE CHECK

1214 SET LISTENER SET PARTICIPANT AS LISTENER, SEND SPEAKER DATA STREAM TO PARTICIPANT IF SPEAKER OR 'NO SPEAKER' DATA STREAM IF NO CURRENT SPEAKER; DISPLAY PARTICIPANT IN ALL GROUP PARTICIPANT LISTS

1215 CONFERENCE CONTROL LOGIC I

FIG. 12a
FIG. 12b
JOIN SUBGROUP?

SET PARTICIPANT AS LISTENER IN
SUBGROUP, SEND CURRENT SPEAKER OR NO
 SPEAKER DATA STREAM TO ATTENDEE, DISPLAY
PARTICIPANT IN SUBGROUP MEMBERS LIST (ALL
SUBGROUP MEMBERS)

IF SPEAKER IN SUBGROUP, SET SUBGROUP TO NO
 SPEAKER AND SEND NO SPEAKER DATA STREAM TO
SUBGROUP; SET PARTICIPANT IN MAIN ROOM, REMOVE
FROM SUBGROUP MEMBER LISTS, STOP DATA STREAM TO:
FROM SUBGROUP TO PARTICIPANT, RECEIVE MAIN ROOM
DATA STREAM OR NO SPEAKER DATA STREAM IF NO SPEAKER

RECONVENE CONFERENCE?

SET ALL PARTICIPANTS INTO MAIN CONFERENCE,
DISPLAY OF SUBGROUPS TO ALL PARTICIPANTS

UNLOCK CONFERENCE?

SUBGROUP BREAKOUT?

SET CONFERENCE LOCK OFF

MOVE SUBGROUP MEMBER?

SUBGROUP BREAKOUT?

MOVE SELECTED SUBGROUP MEMBER TO SPECIFIED
SUBGROUP, REDISPLAY NEW GROUP CONFIGURATION TO
ALL SUBGROUPS

CLEAR HANDS RAISED?

CLEAR ALL RAISED HANDS IN CURRENT
GROUP OR SUBGROUP

CONFERENCE CONTROL LOGIC 4

FIG. 12d
Figure 13a: Alternate Interface Embodiment

Figure 13b: Alternate Embodiment Interface Settings Panel
VIDEO CONFERENCE EVENT SIGN-IN FLOW DIAGRAM

FIG. 14a
REPORT MEETING STARTS IN N MINUTES

YES

REPORT MEETING STARTED N MINUTES AGO

NO

EVENT STARTED IN PAST?

EVENT STARTED WITHIN 15 MIN?

REPORT MEETING BEGINS IN N DAYS/HOURS/MINUTES. PLEASE CHECK EVENT AGAIN 15 MINUTES BEFORE START

REPORT MEETING STARTS IN N MINUTES

SAVE EC AS SESSION VAR

REPLACE WELCOME MESSAGE, RELABEL BUTTON

USER CLICKS EVENT SIGN-IN BUTTON OR JavaScript CLICKS AUTOMATICALLY AFTER 5 SEC

JavaScript LAUNCH CUSTOM SIZED REGISTRATION WINDOW

RELOAD SIGN-IN PAGE WITH EC IN TEXT BOX (RESTORE WELCOME MESSAGE AND ORIGINAL BUTTON LABEL)

REGISTRATION

VIDEO CONFERENCE EVENT SIGN-IN FLOW DIAGRAM

FIG. 14b
Figure 14c: Video Conference Event Sign-In Flow Diagram

1430 Remote Registration
1431: Partner Site Registration and e-commerce (if desired)
1432: Generate Registration Code (RC)
1433: JavaScript launch custom sized Registration Window using Conference site Page URL with RC & Partner Code (PC) in QueryString

1434: Retrieve RC, PC from QueryString
1435: Post PC back to Partner Site using Validate URL associated with Partner Account
1436: Read XML Response: Name, EC

No
1437: Is PC, EC Valid?

Yes
1438: Display Access Violation Message

1439 End

A
REGISTRATION

AID = ATTENDEE ID

AID COOKED EXISTS?

NO

YES

IS AID FOR THIS EC?

NO

YES

RETRIEVE REGISTRATION REQUIREMENTS FOR THIS EC

PRESENT REGISTRATION FORM W/ REQUIRED FIELDS IN CUSTOM WINDOW

USER FILLS FORM AND CLICKS SUBMIT

REPORT ERROR

NO

YES

FORM DATA VALID?

NO

YES

EMAIL REQUIRED?

NO

YES

REPORT "EMAIL SENT. CLICK ON VALIDATION LINK TO VALIDATE YOUR EMAIL ADDRESS, THEN CLICK CONTINUE"

VALIDATION RECEIVED?

NO

YES

REGISTER PARTICIPANT WITH DATA FIELDS, ASSIGN AID, SAVE AID COOKIE

LAUNCH VIDEO CONFERENCE CLIENT IN CUSTOM WINDOW USING EC, AID, DATA ACCESS URL, AND ANY USER SETTINGS

CONNECT TO SERVER

NOTES TO USER: ALLOW A FEW MINUTES BETWEEN CLICKING ON THE EMAIL AND CLICKING ON THE CONTINUE BUTTON. KEEP TRYING UNTIL YOUR EMAIL HAS BEEN VALIDATED.

USE ACCOUNT EMAIL

SEND EMAIL WITH VALIDATION LINK

FIG. 15
MOBILE/PARTNER SIGN-IN/REGISTRATION

PARTICIPANT ENTERS EC

PARTICIPANT Optionally ENTERS UID AND PASSWORD

REPORT INVALID CODE

EC, UID, PASSWORD POSTED TO WEB APPLICATION

IS EC VALID?

UID/PASSWORD?

VALID LOGIN?

EXTRACT AVAILABLE REQUIRED REGISTRATION INFO FROM PARTICIPANT ACCOUNT

RETURN EVENT STATUS, REQUIRED REGISTRATION INFO

REPORT STATUS REQUEST REGISTRATION INFO

USER ENTERS REQUIRED INFO, SELECTS/CLICKS SUBMIT

EMAIL REQUIRED?

EMAIL ADDRESS VALIDATION

NO

BOX OFFICE PURCHASE?

YES

E-COMMERCE PROCESS

POST E-COMMERCE TRANSACTION INFO (IF ANY) ALONG WITH REQUIRED REGISTRATION INFORMATION TO WEB APPLICATION

WEB APP CREATES PARTICIPANT RECORD FOR EVENT, GENERATES AID

WEB APP RETURNS AID, DATA ACCESS URL

MOBILE APP OR PARTNER SITELaunches VIDEO CONFERENCING CLIENT USING EC, AID AND DATA ACCESS URL

CONNECT TO SERVER

MOBILE-PARTNER SIGN-IN & REGISTRATION

FIG. 16
EMBEDDED CONFERENCE

RECEIVE OPTIONAL EC, UID, PASSWORD, REGISTRATION INFO

REPORT ERROR

PARTICIPANT ENTERS REQUIRED INFORMATION THAT IS NOT RECEIVED

REPORT ERROR

EC, UID, PASSWORD POSTED TO WEB APPLICATION

WEB APP RETURNS ERROR CODE

NO

IS EC VALID?

NO

UID/PASSWORD?

YES

VALID LOGIN?

YES

EXTRACT AVAILABLE REQUIRED REGISTRATION INFO FROM PARTICIPANT ACCOUNT

RETURN EVENT STATUS, REQUIRED REGISTRATION INFO

REPORT STATUS REQUEST REGISTRATION INFO*

NO

EMAIL REQUIRED?

NO

BOX OFFICE PURCHASE?

YES

POST E-COMMERCE TRANSACTION INFO (IF ANY) ALONG WITH REQUIRED REGISTRATION INFORMATION TO WEB APPLICATION

WEB APP CREATES ATTENDEE RECORD FOR EVENT, GENERATES AID

WEB APP RETURNS AID, DATA ACCESS URL

EMBEDDED APP LAUNCHES VIDEO CONFERENCING CLIENT USING ED. AID. DATA ACCESS URL

CONNECT TO SERVER

EMBEDDED CONFERENCE SIGN-IN & REGISTRATION

FIG. 17
FIGURE 18 - CLIENT CONNECT TO SERVER

1800: CONNECT TO SERVER

1801: Request Server Connection with user id, conference id, user name, facilitator, record, hw setup

1802: Is this the first user connecting?

1803: Is Conference Valid?

1804: Increase User Count

1805: Update All Clients with New User Info

1806: Does User Already Exist?

1807: Return Client Accepted

1808: Receive END Continue Client Accepted Operation

1809: Return Client Rejected

1810: Receive Client Rejected

END Stop Operation

1811: Delete Old User Information
FIGURE 19 - CLIENT MANAGE CONNECTION

1900: Client Manage Connection

1901: check_in_tries = 0

1902: Wait 2 Seconds

1903: Is check_in_tries > 2

YES

1908: Display dialog informing client of Reconnection Attempt

NO

1904: Increment check_in_tries

1905: Ping Server

1906: Set last_check_in_time to current time for this client

1909: check_in_tries = 0

1907: Return success

FIGURE 18 - Client Connect to Server
FIGURE 20 - Server Manage Connections

2000: Server Manage Connections

2001: Wait 5 Seconds

2002: Get client list

2003: Get First Client Off List

2004: Has client checked in within last 5 seconds?

2005: Are there any more clients?

2006: Get Next Client

2007: Disconnect client

Streaming Server
FIGURE 21 - VALIDATE CONFERENCE

2100: Validate Conference

2101: Set Conference Valid to 'YES'

2102: Request Validation of Conference with conference id

2103: Is Conference ID Valid?

2104: Is startime <= currenttime <= endtime?

2105: Return Validation Response

2106: Is Conference Valid?

2107: Set Conference Valid to 'YES'

2108: Disconnect All Users

2109: Set Conference Valid to 'No'

2110: Set Conference Valid to 'No'

2111: Wait 3 Seconds

Streaming Server

Eduplex Server
Figure 22: Control Stream Data Flow 1

Active A/V Stream

Inactive A/V Stream

Control Status
Figure 23: Control Stream Data Flow 2

Active A/V Stream

Inactive A/V Stream

Control Status

- Attendee 3 Listening
- Attendee 1 Speaking
- Attendee 2 Speaking
- Attendee 1 Listening
- Attendee 2 Listening
Figure 24: Full Color Embodiment

Eric: We sold 1,000 courses last week.
Laurence: Congrats!
Dave: I'm blogging as we speak.
Peter: What's for lunch?
Figure 25: Settings Panel

Settings Panel
(Displays when clicking on Settings button)

Connection Controls
- Reset Audio/Video
- Reset Network

Bandwidth Selection
- High Bandwidth
- Medium Bandwidth
- Low Bandwidth

Other Tools
- Enable Message Alert
- Enable Text Chat
- Record Conference
- Request Extension
- Play Intro Video

(Additional buttons not shown include Play Presentation button)
FIGURE 26 - Initiate Floor Change

2600: INITIATE FLOOR CHANGE

2601: Client Clicks Action Button

2602: Is Floor State = "LISTEN"
   Yes → FIGURE 27: RAISE HAND / TAKE FLOOR
   No

2603: Is Floor State = "HAND RAISED"
   Yes → FIGURE 28: LOWER HAND
   No

FIGURE 29: RELEASE FLOOR
FIGURE 27 - Raise Hand / Take Floor

2700: RAISE HAND / TAKE FLOOR

2701: Request Floor from Streaming Server

2702: Is Floor Open?

YES

2703: Does Client Have Camera and/or Mic?

YES

2704: Set Client Status to FLOOR and Floor Owner to Client

2705: Update All Client of Status Change

2706: Send Floor Received to Client

2707: Client Receives "Floor Received" Message

2708: RECEIVED FLOOR END

NO

2709: Set Client Status to HAND RAISED

2710: Increase Hand Count

2711: Update All Clients of Status Change

2712: Send Hand Raised to Client

2713: Client Receives "Hand Raised" Message

2714: HAND RAISED END

Client

Streaming Server
FIGURE 28 - Lower Hand

2800: LOWER HAND

2801: Send Lower Hand Request to Streaming Server

2802: Does Client Have Hand Up?

2803: Set Client State = LISTEN

2804: Reduce Total Hand Count

2805: Update Clients of Change

2806: Send Hand Lowered to Client

2807: Client Receives "Hand Lowered" Message

2808: END

2809: END
FIGURE 29 - Release Floor

2900: RELEASE FLOOR

2901: Stop Sending Stream to Server

2902: Wait 1 Sec

2903: Send Release Floor to Streaming Server

2904: Does Client Have Floor?

2905: Set Floor Owner to Empty

2906: Set Client State to LISTEN

2907: Update Clients of Change

2908: Send Floor Released to Client

2909: Client Receives "Floor Released" Message

2910: END

2911: END

2912: NEXT SPEAKER
FIGURE 30a - Next Speaker

3000: NEXT SPEAKER

3001: Get Client List

3002: Get First Client Off of List

3003: Is Client's Hand Raised?
- Yes: 3004: Is This the First Hand Found?
  - Yes: Set First_Hand to This Client
  - No: NS-1
- No: NS-2

3006: Are There Anymore Clients?
- No: NS-1
- Yes: 3007: Get Next Client

3008: Set First_Hand to This Client
FIGURE 30b - Next Speaker

3009: Is First_Hand Set?

Yes

3010: Set Floor Owner to First_Hand Client

3011: Decrement Hand Count

3012: Set First_Hand Client's State to FLOOR

3013: Send Floor Received to First_Hand Client

3014: Client Receives "Floor Received" Message

3015: END FLOOR RECEIVED

No

3016: Send FLOOROPEN Message to All Clients

3017: END OPEN FLOOR
SYSTEM AND METHOD FOR INTERACTIVE VIDEO CONFERENCING

I. FIELD OF THE INVENTION

[0001] This invention relates generally to person-to-person and person-to-group conferencing, using video and audio over the Internet. This technology is referred to as video conferencing.

II. BACKGROUND OF THE INVENTION

[0002] Growth of Video Conferencing

[0003] Video conferencing over the Internet is growing more and more popular, especially as the processing power of available Internet devices such as desktop, netbook and laptop computers, smartphones and tablets, etc., increases, and even more importantly, as available Internet bandwidth to the Internet device, either wired or wireless, increases. It is clear that this trend for use of video for business or social interaction on the Internet will continue to increase over the foreseeable future.

[0004] Current implementations of video conferencing fall into several categories: A) webcasting (one-to-many), B) video phone (one-to-one), and C) multiple person simultaneous video conference (MPSVC).

[0005] A) Webcasting (One-to-Many)

[0006] The first category is referred to as webcasting. In this case, the video is one way, streaming from a central location to multiple receivers or participants. Typically, feedback from the participants is limited to text via a text chat function, and in some cases the ability of the student to raise their hand to ask a text question by clicking on a button. This approach is effective for business presentations to prospective clients or customers, as well as for online lectures. This implementation is sometimes offered in conjunction with a teleconference, where all participants are typically muted during the presentation, and are allowed to ask questions via the teleconference at specific points, or at any point in the conference. Teleconferencing is typically used instead of audio over the Internet because the technical challenges of using VOIP (Voice over Internet Protocol) for multiple simultaneous callers can result in a number of problems which can cause a business presentation to end up in chaos, such as problems with feedback and echo, poor audio level, lack of the required hardware on a participant’s computer, multiple people talking at the same time, etc.

[0007] B) Video Phone (One-to-One)

[0008] The second category is a one-to-one videoconference (aka video phone), such as offered by Skype® or FaceTime® from Apple. In this case, two individuals can talk and see each other, assuming both have webcams with built-in or external microphones. Communication in this case is bi-directional, and in some cases (with use of speakers rather than headphones) echo problems will occur, causing some difficulty. An easy solution is the use of earphones, but often the user does not have earphones available. In this case, they must reduce speaker volume, which often makes it hard to hear the other person. Some systems also exhibit sufficient delay to cause talk-over, even with only two people talking. Another solution that reduces this problem is echo cancellation done in software or hardware.

[0009] C) Multiple Person Simultaneous Video Conference (MPSVC)

[0010] Finally, the third category (growing in popularity), is the multiple person simultaneous videoconference (MPSVC). Here, more than two people are in a video meeting. For example, ooVoo.com offers a service that supports up to 6 people at once. Other services allow 8 or more. All of the participants can see the other participants, as well as hear them. This type of service can be utilized by a group of friends in a free-for-all discussion or meeting, or alternatively for an interactive conference. An interactive conference allows a facilitator as well as other participants to speak, hear, and be seen by the others. A facilitator (or moderator or leader) is generally a participant with extra powers.

[0011] This MPSVC category has many problems. First, it may require a very high processing performance and Internet bandwidth, as each participant must receive a separate video stream from each of the participants. There are two types of MPSVC: peer-to-peer, and central server. Alternatively, very small, low quality videos at a low frame rate may be used.

[0012] For peer-to-peer MPSVC conferencing, where no central server is involved, a group of 8 participants requires that every individual participant support 7 outgoing video streams (one to each of the other peers, all the same), and 7 incoming video streams (one distinct stream from each of the other peers).

[0013] While the 7 outgoing streams are just rebroadcasts (copies) of the same stream, and thus only require being generated once by the processor, the total load on the CPU is 8 distinct streams (7 distinct incoming plus 1 distinct outgoing), there are 14 total streams on the Internet.

[0014] For central server MPSVC conferencing, typically the server receives a single incoming stream from each of the participants and resends it to all other participants. Thus, the central server receives 8 distinct streams (one from each participant), and transmits 7 outgoing streams (one from each participant).

[0015] This central server conferencing reduces the load on an individual participant computer to 8 incoming streams from the central server, and 1 outgoing stream to the central server, in the example being discussed. Thus, the central server conferencing has some scaling up advantages over the peer-to-peer conferencing.

[0016] However, in both the central server and the peer-to-peer conferencing, there are serious loads on both the participant’s computer and Internet connection. These loads (in combination with the loads on the central server if used) contribute to a noticeable delay between when a participant speaks and when they are heard. The process of compressing and decompressing the video streams also contributes to the noticeable delay between the original speech and when it is heard at the destination, as does the time to transmit the audio/video stream over the internet.

[0017] In addition to the enormous load on the Internet connection and computers, the audio echo problem is dramatically increased in MPSVC conferencing. Rather than just one set of speakers and microphone (for a webcast), there are eight sets of them, each one capable of generating echoes. The time delays between when someone speaks and when the other participants hear what they are saying (typically longer for a central server solution) results in many “start-stop” conversations.

[0018] “Start-stop” conversations occur when participants hear a dead space (a silent pause), and start speaking. It can be more than a full second before a speaker realizes that a second speaker also started speaking at around the same time. Then
they both stop speaking. Neither of the speakers realizes this for another second or so. Thus, the cycle continues, until one decides to not speak, or tells the other to go ahead.

[0019] The combination of: a) dropped video streams from bandwidth overload, b) computer crashes from computer overload, c) echo problems causing interference with voices, and d) delays causing stop-start conversations, creates a less than stellar experience of having a multi-person simultaneous videoconference (MPSVC).

[0020] The frustration caused by these problems often leads to reverting to the tried-and-true teleconference, wherein dropouts and delays are reduced, and wherein echo problems have been solved (start-stop conversations and talk-over is still a problem in this case). While the multi-person simultaneous videoconference (MPSVC) may be feasible for chats with a few friends, it is not at all acceptable for serious use in business or for online interactive seminars.

[0021] Thus, very expensive hard-wired solutions are typically used in business situations to reduce these problems. These hard-wired solutions requiring elaborate conference room setups at all locations of the conference. While this is an option for Fortune 500 companies, it is not a viable approach for most small businesses or non-profit organizations.

[0022] Conferencing Summary

[0023] In summary, one-on-one video conferencing solutions (videophone) or one-to-many (webcast) video conferencing solutions exist and work well. In contrast, multiple participant (MPSVC) solutions are frustrating and difficult to use successfully, with the possible exception of high cost hardwired elaborate conference room setups.

[0024] Other Interactions such as Subgroups

[0025] There are a number of types of interactions that are common in physical conferences that are not available currently for online conferences. For example, a very useful and often used approach in physical conferences or seminars is the ability to break the conference up into small groups (subgroups) for discussions or exercises. Once the discussion or exercise is completed, the conference is reassigned by the leader or facilitator into a single meeting. This may occur multiple times during a single conference, either with assigned subgroups or ad hoc subgroups. Subgroups may be static or dynamic: the meeting facilitator may request that subgroups stay together once formed for the duration of the conference (static subgroups), or may request that a new set of ad hoc subgroups be formed for each break-out (dynamic subgroups). Also, it is common for the facilitator to move around and listen in on subgroup proceedings, and in some cases interject comments to help the subgroup process along.

[0026] Multiple participant conferences can be single events, or can be a series of events over a period of time, such as part of an extended course or training. In some cases, static subgroups are maintained over multiple sessions or conferences within a larger context, such as an educational training course. In other cases, each session or conference creates a new set of subgroups. There are no known videoconference systems available that offers full dynamic and static subgroup support, including for extended, multi-event settings.

[0027] Another important issue is that most video conferencing systems are complex, and often intimidate non-expert users. This results in lost time while users are trying to figure out how to use the system, and often requires coaching via telephone to help new users get onto a conference.

[0028] What is Needed

[0029] What is needed is an easy to use mechanism for providing interactive video conferencing for a group of people that reduces the CPU and Internet bandwidth load, and addresses the echo problems and start-stop delay problems described above, while still allowing for each participant to speak, be heard, and be seen.

[0030] Secondly, what is needed is a system and method for rapid switching between speakers, automatically and easily controlled via a simple and easy to understand interface, thus allowing the system to automatically control conversation flow, and enhance interactivity. This approach is called Automatic Flow Control Video Conferencing (AFCV). The system also should provide a way for manual override of conversation flow by a facilitator.

[0031] Thirdly, what is needed is a mechanism for creating subgroups within an interactive videoconference to enable effective discussions and small group exercises. Additionally, what is needed is a mechanism to enable group size to be independent of the load on a participant’s Internet device and Internet connection, thus allowing any number of participants, only limited by the central server’s ability to support conference “seats” or video streams. Additionally, what is needed is a system and method for managing both static and dynamic subgroups, both within a single event and in an extended multi-event. Finally, what is needed is an interactive video conferencing system that supports at least two methods for conference control: a facilitator central control over who has the floor, and group control of who has the floor. The latter is most suited to operation within subgroups or ad hoc conferences.

III. SUMMARY OF THE INVENTION

[0032] Improvements

[0033] The teachings herein improve over conventional techniques by modeling the complex relationship information of an operation of an entity.

[0034] In accordance with the present invention, a system and method is disclosed for providing a video conferencing system that eliminates the echo problem (and its attendant earphone requirement), reduces each participant’s CPU and Internet bandwidth load to a single incoming or outgoing video stream no matter how many participants are in the conference, and provides for the creation, use, and disposal of static or dynamic subgroups during an ongoing conference, including the recreation of static subgroups in multi-event cases. Additionally, the present invention eliminates the problem of start-stop conversations (due to delays between the speaker and the listener brought on by server, Internet, and processing delays). Finally, the present invention allows for both facilitator and group control of the conference in an easy to use manner.

[0035] Class Mode, Group Mode, and Multi-Interactive Mode

[0036] In one illustrative embodiment with a facilitator, the facilitator controls who is speaking at any given time. This is referred to as Class Mode.

[0037] A second mode of the illustrative embodiment provides for shared or group control of which participant is speaking. This is referred to as Group Mode. In Group Mode, each participant has less control over who is speaking at any given time (as compared to a facilitator in Class Mode).

[0038] A third mode, referred to as Multi-Interactive Mode, allows a facilitator to break up a conference into subgroups for the purpose of discussions or exercises. In this third mode,
one or more facilitators can visit any subgroup to listen in or interact. Alternatively, one or more of the subgroups may be private, and exclude the facilitator from listening or interacting.

[0039] The various modes of operation will be described in greater detail below. The present invention permits one person at a time to speak, or have the floor, per group or subgroup. This permission may be rapidly switched between participants under the control of a facilitator, or under control of the participants, or both. Conversations within groups are private to each group.

[0040] The present invention allows participants to connect to the conference using: a webcam and keyboard (audio, video and text input), or just a microphone and keyboard (audio and text input), or with a keyboard (just text input). In the latter case, the participant with only a keyboard cannot take the floor to audibly speak because they do not have the hardware (microphone) required for that purpose.

[0041] However, a participant with only a keyboard can send text messages (either comments or questions) to the facilitator (and/or optionally to other participants), as described below. In one embodiment, a text-only participant can also “raise their hand” using a graphical user interface in response to a question or poll, but if selected by the current speaker, the floor will not be given to them. Instead, their “hand raised” status will be simply cleared, and if they also have a message pending then the message may be displayed to all participants. In other words, raising the hand for a text participant can only be used for responding to a poll, vote, or other form of signaling, and does not represent a request to take the floor and audibly speak. This difference may be indicated by a different graphic than used for a participant that can take the floor and raises their hand, such as a question mark or other graphic.

[0042] Special Cases (Dear Participants)

[0043] Further, these procedures may be modified for special cases (such as deaf participants), so that “taking the floor” by a participant with only a keyboard may result in text from a selected participant being continuously displayed in an image window (similar to a video image). Additionally, audio inputs from a deaf participant may be deleted or blocked, in order to prevent undesired background noise of which the deaf participant may be unaware. Blind participants may not need incoming visual streams. Additionally, word recognition software may provide continuous text to accompany the video image (and optional audio sound) of a speaker (so that deaf participants can read along, and/or to provide a text record of the conference). Although not yet commercially available, gesture recognition software may soon provide continuous text (or machine generated audio) to accompany the video of a deaf speaker who is “speaking” in sign language (such as American Standard Sign Language). The point is that a person of ordinary skill in the art will, based upon the present disclosure, understand configurations and procedures that are not expressly stated in the specification.

[0044] Facilitated Conferences

[0045] For facilitated conferences, the facilitator (or moderator, or leader) can close the conference to new participants at any time (lock the door). This feature is useful when a series of planned interactions require the participant to be present from some point forward in the conference, and having new participants join the conference after that point would be distracting and require covering ground multiple times, thus punishing participants that arrived on time. In other words, the facilitator can lock the door against new participants.

[0046] As described above, conference participants may indicate the desire to speak by raising their hand. Participants may also submit text messages (questions or comments) at any time, even if they also have video and/or audio capability. All participants may see a list of conference participants, as well as their status, such as if they are requesting permission to speak (raised hand) or have submitted a text message (message pending). Alternatively, text messages are automatically displayed in chat format. The facilitator may block such chat traffic if desired. A request to speak (hand raised condition) is generated by a control device, such as touching a button on a touchscreen, clicking on a button (in a graphical user interface), tapping on a specific key (or combination of keys) on a keyboard, or other control apparent to one skilled in the art.

[0047] As described above, the raising of hands is also a useful polling or voting mechanism, such as the facilitator asking “How many did the homework?” A way for the facilitator to clear all raised hands in a single operation is provided in one embodiment, which facilitates additional votes, and which removes the burden on each participant to actively lower his hand after voting. Alternatively (or additionally), hands may be automatically lowered after a given period of time, such as 20 seconds.

[0048] The current speaker (the participant on the floor) can grant the floor, i.e., give another participant the opportunity to speak, to any participant with the exception of a text-only participant as described above. Alternatively, the current speaker may relinquish the floor by clicking on or touching a control such as a button or icon or by pressing a key or combination of keys. In this case, where one or more attendees have raised hands, the present invention will automatically transfer the floor to the first person that raised their hand (automatic flow control).

[0049] Granting permission to speak (transferring the floor) is thus performed by using a control such as touching or clicking on the selected participant in the participant list or by releasing or relinquishing the floor. Revoking permission, or taking back the floor, is performed by a facilitator using a control such as clicking or touching their own icon or name in the participant list, or other manners of control. Generally, only a facilitator has this ability to force termination of another participant’s position on the floor.

[0050] Additionally, a participant that has the floor can return the floor back to the facilitator by a control device such as clicking or touching a button, pressing the space bar or by some other control. When multiple facilitators are present, the floor may be returned to the facilitator that most recently had the floor, or to the facilitator that set up the meeting (the owner). Alternatively, releasing the floor does not transfer the floor, rather, additional action must be taken by a participant to take the open floor. In a non-facilitated conference, this same action will release the floor without transferring the floor to another participant, unless one or more hands are raised (requests to speak). In that case, the first participant that raised their hand will receive the floor. Various control devices for these functions will be apparent to one skilled in the art. For example, in a non-facilitated conference including tenured teachers, the teacher with the longest tenure that has their hand raised may receive the floor. Alternatively, participants may bid money (through a credit card or similar) for the
opportunity to speak next on the floor, where the highest bid receives the floor. The bid money may be donated to a charity.

[0051] As described above, multiple facilitators are also supported in the preferred embodiment. In one embodiment, all facilitators can assert the floor at any time, including taking the floor from another facilitator. Naturally, this feature should be used with restraint. This is referred to the "play nice" rule for facilitators.

[0052] An alternate embodiment allows only one facilitator to have full control at one time (and the controlling facilitator may transfer the control to another facilitator). Or, somewhat similar to the tenured teachers, the facilitators may have a hierarchy of control wherein higher facilitators may take the floor from lower facilitators, but not the reverse.

[0053] The proposed system generally has a single video stream per participant Internet device, and that single video stream is either incoming (the participant is viewing and hearing another participant, or the facilitator, or a notification or instructional video, or similar), or outgoing (the participant has the floor, and is speaking and being viewed by other participants, or is playing a video for the other participants). The system also supports participants with audio input only (no webcam), wherein a photo of the participant, a placeholder icon, or a notification video may be displayed rather than a video of the participant speaking. Alternatively, if the current speaker is audio-only, a graphic "Audio Only" is displayed in place of the video. A "Floor Open" graphic may be displayed when no one has the floor.

[0054] In one embodiment, the central server logic may be simplified by sending the speaker video stream to all participants (including the speaker). In this case, the speaking participant may use the speaker video stream from the central server for visual feedback (very useful), or may verify that the central server is properly sending the speaker video stream showing his face (verify that he is "on TV"), or may ignore the speaker video stream from the central server. In this case, only the speaking participant would have two streams (his speaking outgoing stream plus the repeated speaker stream from the central server), and all other participants would have a single stream only, as discussed above.

[0055] Even if the central server does not send the speaker video stream to all participants, the speaker may still receive visual feedback by monitoring his own outgoing speaker video stream. This is the preferred embodiment, because of the potential for confusion from the round-trip delay in the video if send to and received back from the server.

[0056] Subgroups

[0057] A Multi-Interactive conference that has broken into subgroups will also have a single video stream per participant Internet device. However, there will be multiple video sources to and multiple video sources from the central server during subgroups. For each subgroup, there will be a distinct video stream from the speaker in each subgroup (or a notification or instructional video) going to the other subgroup members in the same subgroup. Thus, the central server will receive one stream for each subgroup (the speaking subgroup member), and will transmit this same stream to the nonspeaking members of the same subgroup.

[0058] Eliminate Problems

[0059] Because the present invention utilizes one or more rapidly redirectable one-way video streams, the echo problems (and the attendant requirement for earphones) and the "start-stop" problems are eliminated completely. The delay when the floor is being transferred from one participant to another actually helps maintain a smooth flow of conversation because it is clear when you are to speak and when you are to listen. Specifically, graphical and other ways may be used to signal to a participant when they have the floor, such as a specific sound ("a rising beep sound") or a change in the displayed user interface (such as a flashing icon in a status panel). The present invention also eliminates talk-over and start-stop conversations, where multiple people are talking (or attempting to talk) at the same time. This is especially problematic for prior art multiple participant videoconferences (MPVSC) during discussions.

[0060] An additional feature of the present invention is the delayed video stream termination feature. This feature overcomes the problem of the differential in speed between the video stream and control signals between the client application and the server. Specifically, the user's action, such as clicking on a user interface element, will be nearly instantaneously transmitted to the server, while outgoing video is delayed by the compression, packing, and protocol management for transmission of the video stream. Thus, if the floor is revoked immediately upon receiving a transfer or release, the last portion of time in the streaming video will be cut off. The delayed video stream termination feature waits a set period of time after receiving the command before terminating the outgoing video stream. Other dynamic methods are available to one skilled in the art, such as watching the incoming data stream and looking for an end before taking action at the server.

[0061] Also, because only a single video stream is being either received or sent from each participant's Internet device at one time, the processing load and Internet bandwidth load for each participant's Internet device is very low (approximately half the load of a one-on-one video conference in the prior art, wherein both an incoming and an outgoing stream are simultaneously required). Additionally, the processing load and bandwidth load for each participant's device is completely independent of the number of participants in the conference.

[0062] Optionally, a listening participant's device may regularly (perhaps once a minute, or once every few seconds) transmit a small image so that a thumbnail image may be displayed and regularly updated on the graphical user interface adjacent to the name of the participant. This kind of data load is minimal, but may be useful for spotting when listeners are falling asleep during a classroom conference, or are not actively participating. An alternate embodiment would be to enable a facilitator to access these images by a control device, such as hovering their cursor over the participant's name, or clicking or touching their name, for example. This feature may only be available to a facilitator, and the images may only be transmitted from a server to the facilitator's Internet device when requested. In another embodiment, the control device may send a request for a thumbnail image when activated by the facilitator. In this case, images may only be transmitted when requested by the facilitator. The drawback to this approach is the slight delay between activating the control device and receiving the image. The advantage is that no bandwidth is taken to send images unless specifically requested. In all cases, the ability of a facilitator to monitor participants is provided without requiring video streams from each participant simultaneously.

[0063] A preferred embodiment also includes a mechanism that manages connection interruptions and failures. This mechanism includes a polling technique, described in more
detail below. This mechanism allows the client applications to detect a disconnect from the server caused by Internet traffic interruption, and automatically reconnect (or attempt to reconnect). Likewise, the server may use this mechanism to automatically detect lost connections to signal that a participant is no longer present, such as removing them from the attendee list, changing the color of the attendee’s name, or other methods known to one of ordinary skill in the art.

[0064] Two Way Video Streams

[0065] Although less efficient, an alternative embodiment may utilize two way video streams. For example, each attendee device may continuously output its outgoing video stream to the server. The server may select the appropriate video stream (the video stream from the speaker with the floor) and send the selected video stream to all of the attendees (or perhaps to all of the attendees except the speaker attendee). In this fashion, each attendee would receive one stream, and send one stream.

[0066] The server would receive many streams, but only needs to process (and send out) the stream of the speaker. This two way video streaming method requires about twice as much bandwidth as the preferred one-way video streaming embodiment.

[0067] This two way video stream embodiment does have some minor advantages, such as not needing overhead (control) for switching on and off the video streams of the attendee devices (which can remain on). Also, this two-way embodiment allows the server to more quickly switch from a first video stream to a second video stream, because the server does not have to send a command to the second attendee device to start sending the second video stream (and does not have to wait for the second attendee device to obey the command). However, the switchover must still take into account the time delay between the received command and the video streaming from the speaker, as well as the time delay of the next speaker receiving and viewing the same video. A switching delay must take into account both delays to avoid transmitting the last few words from the speaker heard via the microphone of the next speaker when switching to their video stream.

[0068] Finally, the server can monitor incoming video streams to detect the loss of connection or other technical problems more quickly, and automatically handle them in some cases. For example, a lost video stream may cause the attendee’s name in the displayed list to turn red, indicating trouble. This approach would replace the polling mechanism of the preferred embodiment, thus simplifying both the server programming and the client programming and complexity.

[0069] However, these advantages generally do not justify the doubled bandwidth required for two way video in comparison to one-way video, and requires substantial increase in bandwidth at the server.

[0070] Downloaded Application

[0071] This present invention may utilize a central video-streaming server that receives the incoming video from the participant or facilitator currently having the floor, and sends the received video data out to the other participants in the conference. If no one has the floor, a “no speaker” notification video loop may be presented in place of a participant video, or a graphic may be displayed including the text Floor Open, for example. This central video-streaming server may be responsible for switching the floor to the appropriate participant (or participants when multiple subgroups are involved) under the direction of control received from facilitators or participants. In one embodiment, participant control and user interface are provided by a downloaded client application from the website providing the conferencing service. This downloaded client application then connects to the central server, and the downloaded client application is then responsive to the user actions and interactions with the central server. In an alternate embodiment, an application may have been previously received or downloaded, and is launched for the purpose of connecting to the videoconference. An example of this would be a video conferencing application purchased in an online store and downloaded to a tablet or smartphone.

[0072] The processing load on the central server is approximately linearly proportional to the number of participants in the conference. This processing load may be controlled by limiting the number of participants (or “conference seats”) that are allowed on the server at any given time as part of the conference scheduling function.

[0073] The downloaded application for use on the participant’s Internet device can be provided using multiple technologies. Typically, however, a CPU-independent technology, such as Adobe Flash® or Sun Microsystems’ Java®, is used to maximize the compatibility of the downloaded application to the various possible Internet devices that might be used by various participants. The alternate embodiment, an installed application, may also be acquired by the Internet device owner and installed in the device prior to the use in a videoconference.

[0074] Dynamic Subgroups

[0075] Within a given conference, subgroups are formed by organizing a set of separate rooms for the subgroups dynamically. Each room has a predetermined number of participants, selected either by the system, by a facilitator, or by ad hoc grouping controlled by a facilitator. Each room becomes a separate (almost independent) videoconference, typically with control mechanisms to support subgroup control of the floor. A facilitator generally has the ability to “visit” any room to hear and see what is going on, and to interject comments.

[0076] Subgroup termination can be: automatic by “time out” based on a predetermined period of time, or by subgroup action, or by facilitator action, or by a combination of the above. Automatic termination may be controlled by a timer countdown optionally visible on all participant user interfaces, and termination may be instant, i.e., cutting off any in-process conversations within the subgroups.

[0077] Alternatively, the timer countdown is for reference only, and the facilitator can post a text message to finish up quickly and return to the main conference. In a preferred embodiment, such messages to the entire conference from a facilitator may be emphasized via the generation of an alert sound (and/or a flashing visual signal). When “finish up” message is received, the subgroups can choose when to return to the main conference. However, a facilitator can also force the return at any time of some or all participants. In either case, a facilitator generally automatically has the floor to continue the main conference. Alternatively, a facilitator takes the floor when they are ready to begin.

[0078] The present invention also supports requests to extend the time for a conference. Extensions generally are granted if sufficient conference seats are available at the central server (if the server has sufficient capacity, and perhaps if the conference participants have sufficient money or has prepaid for seat time, or has an unlimited meeting account). In a preferred embodiment, only the facilitator has this capability.
for extending the time of the conference. In an alternate embodiment, all participants have the ability to request an extension of the conference.

[0079] Options for Subgroup Formation

[0080] A facilitator generally has multiple options for subgroup formation, selectable in the facilitator-user interface. The size and composition of subgroups will depend on what the facilitator is asking the subgroup to do. For example, some small group exercises work best with 2 participants (dyads), others with 3 participants (triads), etc. The present invention also supports the dynamic configuration of dyads, called a mill. A mill is a series of dyad interactions, where each participant interacts with as many other participants as time allows. In a mill, dyads are generally formed randomly by the system. Alternatively dyads may be formed in some non-random fashion. For example, in a dating mill, the dyads may be formed by an artificial intelligence engine that determines the dyads based on participant compatibility, such as by sex, age, or other factors.

[0081] Interactions within the dyad are controlled by the participants, and termination of the dyad is also controlled by the participants, by mutual agreement, as described below. Alternatively, dyads may be terminated by a single participant (effectively “leaving the room” or “ditching the date”), or by a predetermined period of time ending, or in other ways.

[0082] When a dyad is terminated within a mill, the system connects each available, unconnected participant to another available participant in a random manner. The system keeps track of which participant has connected with which other participant, so subsequent dyads are always “first time” with the other participant for the current mill. A mill is terminated by a timer countdown, or by facilitator action, or by a lack of new dyads, or in other ways. Text (or audio or graphic) warnings to finish up are also supported, as with normal subgroups.

[0083] A preferred embodiment automatically forms subgroups randomly, and allows facilitators to reconfigure the groups by a control, such as by dragging participant icons from one subgroup to another. Other control approaches will be apparent to one skilled in the art. The system may be set to automatically retain subgroups (static subgroup formation), or to create new subgroups each time that the main group breaks up within a given conference (dynamic subgroup formation) or within a related set of conferences. In a preferred embodiment, this static or dynamic condition is set as a preference by the facilitator when scheduling the conference. Also set by preference is the option to maintain the same subgroups in subsequent conferences with the same participants (for example, in the case of weekly conferences for a specific subject such as a college class, or for regular religious activities). Subgroups, once formed, can be reused within a single conference, or can be replaced with new subgroups of the same or different size, under the control of a facilitator. Subgroup information for static groups is maintained in a database, along with other information that controls the use of the conferencing system.

[0084] Static subgroups can be maintained while still allowing dynamic groups for specific occasions. This is a valuable tool when a conference facilitator wishes to create “home” groups that will work together throughout the conference, but allow dynamic groups for certain exercises, such as for a mill, or for certain purposes. The system may also support multiple types of static groups, and allow a facilitator to name them and choose between them and dynamic groups for each breakout session. Static groups also may be pre-defined and named prior to the conference or conference series by facilitators, and can be edited by a facilitator at any time.

[0085] Setting up and using subgroups requires a first step wherein a facilitator initiates the formation of the subgroups, while the full group is still present in the main conference room. The facilitators and other participants can view the subgroups in the provided attendee lists, allowing for adjustments for special cases by a facilitator, such as not wanting spouses to be in the same subgroup. Group auto-formation can also be controlled by age, sex, or any other item of information available from participant profiles. This is referred to as profile-based subgroup formation. This profile-based logic may also be applied to dyad formation.

[0086] Regarding terminology, a “facilitator” is generally a participant with special facilitation powers. An attendee is generally an ordinary participant without facilitation powers. Thus, there must be at least two participants for the system to function. However, generally the participants comprise at least one facilitator and at least two attendees.

[0087] Once satisfactory subgroups are formed, and the facilitator has instructed the participants on what they are supposed to do in their subgroups, then the facilitator commands the system to instantiate the subgroups (subgroup break-out) using a control, and subgroup interactions begin. If desired, the facilitator may specify the grouping as a static group, and assign a name so it can be invoked again in the future. The subgroup generally begins with no speaker selected. Alternatively, the system or the facilitator can select the first speaker. Each subgroup may have a sub-facilitator, which may be selected randomly, by the facilitator, or by some other logic such as profile based, or by agreement between the subgroup members.

[0088] Maintaining Static Groups Across Multiple Conferences

[0089] As stated earlier, a dynamic grouping can be saved for future use, and assigned a name. Within a given conference, a facilitator can reform this same group configuration by selecting it. This allows them, for example, to form static “home” groups that are reused multiple times, interspersed with dynamic and/or other static configurations.

[0090] In order to support static groups across multiple conferences, a Repeat Conference function is provided. This can be used with or without static groups, and allows a continuing meeting, such as a weekly staff meeting to use the same event code (or a similar event code, such as with a changed suffix). When repeating a conference, it copies all of the settings from the conference into the new conference, but moves the start date up one week. The start date can of course be changed, as desired. This repeat conference function is available from a Conference or Event History, which lists all conferences scheduled by the owner, including future events. Thus, a conference owner can set up a series of meetings all in the future, or set up the next meeting after the current one has completed.

[0091] If a series of conferences are set up using the same unique event code, all static group definitions saved will be available to each successive conference. Of course once loaded, they can be modified and resaved under the same or different name, as needed. Thus, static groups can persist over multiple conferences.

[0092] Additional Advantages

[0093] Additional advantages and novel features will be set forth in part in the description which follows, and in part will
become apparent to those skilled in the art upon examination of the following and the accompanying drawings or may be learned by production or operation of the examples. The advantages of the present teachings may be realized and attained by practice or use of the methodologies, instrumentalities and combinations particularly pointed out in the appended claims.

IV. BRIEF DESCRIPTION OF THE DRAWINGS

 Fig. 1 is a block diagram of an interactive video conferencing system.

 Fig. 2 is a block diagram of the software on the user Internet device.

 Fig. 3 is a block diagram of the software stack on the web server.

 Fig. 4 is a block diagram of the software on the video-streaming server.

 Fig. 5 is a facilitator user interface, with a facilitator speaking to the attendees.

 Fig. 6 is the facilitator user interface, with the facilitator listening to a student.

 Fig. 7 is an attendee user interface, with a hand raised.

 Fig. 8 is an attendee user interface, with a message pending.

 Fig. 9 is an attendee user interface, with a message displayed.

 Fig. 10 illustrates multiple conference modes.

 Fig. 11 illustrates multiple action commands.

 Fig. 12a illustrates a first portion of a conference start request diagram.

 Fig. 12b illustrates a second portion of the conference start request diagram.

 Fig. 12c illustrates a third portion of the conference start request diagram.

 Fig. 12d illustrates a fourth portion of the conference start request diagram.

 Fig. 13a illustrates a graphical user interface, with one hand counted.

 Fig. 13b illustrates another graphical user interface.

 Fig. 14a is a first portion of a sign-in diagram.

 Fig. 14b is a second portion of a sign-in diagram.

 Fig. 14c is a third portion of a sign-in diagram.

 Fig. 15 is a registration diagram.

 Fig. 16 is a mobile/partner sign-in/registration.

 Fig. 17 is an embedded conference.

 Fig. 18 is a client connecting to server.

 Fig. 19 is a client management connection diagram.

 Fig. 20 is a server management connection diagram.

 Fig. 21 is a conference validation diagram.

 Fig. 22 shows a first and second audio/video stream connection.

 Fig. 23 shows a third and fourth audio/video stream connection.

 Fig. 24 shows a take floor interface.

 Fig. 25 is a settings panel.

 Fig. 26 is a diagram to initiate a floor change.

 Fig. 27 is a diagram to raise hand and take floor.

 Fig. 28 is a diagram to lower hand.

 Fig. 29 is a diagram to release the floor.

 Fig. 30a shows a first portion of going to the next speaker.

 Fig. 30b shows a second portion of going to the next speaker.

 V. DETAILED DESCRIPTION OF THE DRAWINGS

 Stable Video Conferencing

 In the following detailed description, numerous specific details are set forth by way of examples in order to provide a thorough understanding of the relevant teachings. However, it should be apparent to those skilled in the art that the present teachings may be practiced without such details. In other instances, well known methods, procedures, components, and circuitry have been described at a relatively high-level, without detail, in order to avoid unnecessarily obscuring aspects of the present teachings.

 Terminology

 The terms “attendee” and “participant” and “user” and “client” are used broadly throughout the specification and claims to include anyone who has connected to a given conference. This includes both facilitators as well as non-facilitators. Depending upon the context (such as in the context of the term “facilitator”), the terms “attendee” and “participant” may be used to indicate non-facilitators.

 The terms “facilitator” and “moderator” are used broadly throughout the specification and claims to include a conference attendee that has additional capabilities, such as the ability to control who has the floor at any time, as well as other controls. Other similar terms include “teacher,” “mentor,” “instructor” for example.

 An attendee can be either a facilitator or not a facilitator. Thus, the term “non-facilitator” is used broadly throughout the specification and claims to include an attendee that is not a facilitator.

 The term “state” is used in the claims to describe, for example a first state wherein a first attendee is selected as a speaker, and a second state wherein a second attendee is selected as a speaker. Transitional states between the first state and the second state may be allowed. For example, releasing the floor transitions from the state of having the floor to the state where local transmission of video has been terminated, to the state where the message that the floor has been released is sent to the server, to the state where the server has terminated the incoming video stream, to the state where the server has sent out the message that the floor is open, to the state where each client application receives the message and displays the new state.

 Additionally, the terms “selecting” and “deselecting” are used broadly. For example, an attendee may desel ect themselves (yield the floor), and the system may enter a transitory state waiting for another attendee to select themselves (take the floor).

 Further, a facilitator attendee may seize the floor by selecting themselves (and automatically deselecting the current speaker). For example, the facilitator may seize the floor by simply clicking a “take floor” button, or by clicking on their own name in the attendee list. Additionally, the facilitator may transfer the floor from the current speaker to another attendee by clicking on the desired attendee name in the list. Typically, this type of control is only available to a facilitator, and not to non-facilitators, and is included in the term “manual conversation flow control.”
Stable Video Conferencing

The present invention relates to delivering stable video conferencing to multiple participants, allowing for subgroup formation and interaction under facilitator observation and control, or alternatively group control of a video conference, while at the same time reducing or eliminating the normal problems of multi-party video conferencing systems, including delays, overload of Internet devices and Internet connections, start-stop conversations, echoes and attendant requirements for use of earphones. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein. For example, the present invention applies to communication systems such as local area network and wireless systems, in addition to Internet communication systems.

The present invention comprises a system and method for providing a highly reliable and stable video conferencing system for an interactive, live conference with multiple participants, and to optionally provide for small group formation for subgroup interactions and exercises within a larger conference. The present invention is specifically designed to overcome the primary hurdles to use of video conferencing for such interactive conferences, including echoes, delays, start-stop conversations, and CPU and Internet bandwidth overload.

FIG. 1 (System)

FIG. 1 is a block diagram of an interactive video conferencing system.

Referring to FIG. 1, the Video Conferencing System Diagram 100 shows multiple participants 108 using various types of Internet devices 104, 105, 106, 107 to connect to the World Wide Web 101 via wired or wireless connections 110. The invention can also be practiced on a local or private network. Internet devices shown are desktop computer 104, laptop computer 105, touch-screen device 106, and Smartphone 107. Other types of Internet devices also can be used, as well. Each of the Internet devices 104-107 are connected to the world Wide Web 101, which is also connected to the web server 102, video streaming server 103, and optionally a database server 110. In an alternate embodiment, the web server 102 and video server 103 functions can be combined on the same server, or can be distributed on any number of servers, using standard techniques well known in the art for configuring servers to handle loads beyond the capacity of a single server. In another alternate embodiment, the database server 110 and web server 102 are combined and run on the same server hardware, and the video server 103 is hosted on a separate hardware platform. Many different configurations are possible and will be apparent to one skilled in the art, but the functionality of a web server 102 and video-streaming server 103 are required.

The functions of the database server 110 can be provided in various ways apparent to one skilled in the art. Alternate embodiments for the database server 110 include configurations where the required information is normally stored in a database and provided via a defined programming interface or protocol in the preferred embodiment is instead stored in temporary memory on the web server 102 or on the Internet device 104-107. Standard storage mechanisms include session variables on the web server 102, cookies on the participant Internet device 104-107, and query string values stored in web page requests sent from the Internet device 104-107 to the web server 102 from information originally delivered to the Internet device 104-107 within a requested web page from the web server 102. Other data storage and retrieval mechanisms are also possible in alternate embodiments of the present invention, and will be apparent to one skilled in the art.

In one preferred embodiment, the web server 102 communicates with the database server 110 in order to provide the integrated functions of the present invention, such as keeping track of scheduled conferences, keeping track of participants, limiting participants based on some criteria, such as participants who have signed up or fulfilled some other prerequisite for the conference, using profile-based subgroup formation, etc. The database server 110 maintains a database of information that can be accessed and modified by the web server 102, and also may include programming code that can be initiated by the web server 102. An example of programming code is Structured Query Language (SQL), available from companies such as Microsoft Corporation of Seattle Washinyon, as well as many others. This programming code can be transmitted to the database server 110 from the web server 102, or can be stored on the database server 110 as stored procedures that can be launched from the web server 102 using communication protocols apparent to one skilled in the art.

In the preferred embodiment, the participant Internet device 104-107 also communicates with the video server 103. This communication is provided via a component of the video conferencing system 100, described below.

In alternate embodiments, other communication modes may be used, such as communication between the participant Internet device 104-107 and the database server 110, and between the video server 103 and the database server 110, or between the web server 102 and the video server 103. The present invention operates within the environment 100 described above, interconnected via the World Wide Web 101 or alternatively, on a private network. The present invention, thus, may execute programming code on the Internet devices 104-107 that is downloaded from the web server 102 interacting with programming code executing on the web server 102, and also interacting with programming code executing on video streaming server 103. In a preferred embodiment, web server 102 also initiates programming code on the database server 110. Each of the components of the present invention will be described in association with FIGS. 2, 3, and 4 below. In an alternate embodiment, programming on the Internet device 104, 105, 106, 107 may be acquired and pre-installed prior to a first videoconference, using standard methods known to one of ordinary skill in the art.

FIG. 2 (Software on Device)

FIG. 2 is a block diagram of the software on the user Internet device.

Referring to FIG. 2, User Internet Device Software/ Hardware Stack 200 shows the configuration of the typical Internet device 104-107. Many other configurations are possible, depending on the cost point for the device, size and
weight, and other considerations. However, all Internet devices have similar functionality as shown in FIG. 2.

Each Internet device 200 is based on specific hardware 201, which typically includes processors and memory, as well as a display device, control devices such as a touch screen or keyboard and function keys or keyboard and mouse, and a webcam and microphone. Other hardware components will also be included, depending on the target market for the device. For example, in the case of the Smartphone 107, a cellular radio transceiver will be included in the hardware. For a desktop computer 104, a hard drive or solid-state drive will be included, along with a wireless or wired Internet connection. Each of these configurations of Internet devices has a minimum configuration including a webcam, a microphone, a processor, memory, and a ability to connect to the Internet via Internet connections 110. Alternate minimum configurations include only a microphone (no webcam) and/or only text entry.

In the preferred embodiment, Internet devices 104-107 will have a well-defined hardware driver layer 202 to separate from the operating system layer 203 functions that are dependent on the specific features and operation of the particular hardware devices included in hardware 201. This is also referred to as the “hardware abstraction layer” in that the interface provided by the hardware driver layer to the Operating System layer 203 above remains constant even when changing to different hardware, matched with its hardware driver.

The operating system (OS) 203 is responsible for handling the many I/O (input/output) services and functions that are typically required by the Internet device. This typically includes routines for displaying graphical user interfaces, processing control inputs from the participant, handling the interactions with the Internet connection, handling audio and video input and output, including video and audio compression and decompression, and adhering to various standards, such as Internet communications protocols and data protocols, such as MP3 for audio, MPEG for video, etc. The OS 203 may use these functions in software, or via a combination of driver and hardware, or even in some cases, entire functions may be implemented in hardware.

The OS 203 provides an Application Programming Interface (API) that is standardized, and typically widely available for programmers to utilize in creating Application Programs to execute on the Internet devices. This standardization is important because many tools and documentation are required to support programmers using the API effectively. Examples of several widely used operating systems include: Windows from Microsoft Corporation in Seattle, Wash., Mac OS X from Apple, Inc. in Cupertino, Calif., and Solaris from Sun Microsystems in Santa Clara, Calif. Smaller devices might use VxWorks from Wind River Corporation of Alameda, Calif., Palm OS from Palm, Inc. of Sunnyvale, Calif., or Android from Google in Mountain View, Calif., or iOS from Apple, Inc. in Cupertino, Calif., or other lightweight operating systems to reduce cost and improve performance. Other alternate embodiments include an integrated application that is designed specifically to execute directly on the hardware 201, or other blending of the various layers of the preferred embodiment software stack shown in FIG. 2, all such variations apparent to one skilled in the art.

In a preferred embodiment, one application that is available to be executed on the OS 203 layer is an Internet browser 204. This application program is capable of connecting to web servers from many organizations and displaying documents according to standard languages, such as HTML, JavaScript, and others. The browser in a preferred embodiment connects to the web server 102 and displays an interface to the user that allows the user to launch the video conferencing application described herein. This operation is typically initiated via a hyperlink or a button on a web page served to the Internet device 104-107 by web server 102.

One of the common functions of a browser 204 is the ability to support plug-ins to expand the function of the browser. One such plug-in utilized by the preferred embodiment is called a Virtual Machine Browser plug-in 205a. An example of this type of plug-in is the Adobe Flash player, from Adobe Corporation of San Jose, Calif. The plug-in provides an Application Programming Interface (API) or a scripting language interpreter or some other standardized way for a custom application to be developed that runs within the plug-in 205a environment.

In a preferred embodiment, the conferencing application 206a is downloaded from the web server 102 in one of several ways apparent to one skilled in the art, and runs within the virtual machine browser plug-in 205a environment. One way to accomplish the download is to embed an object in the HTML code of a web page delivered to the Internet device 104-107 by the web server 102. Another way is for script programming embedded within the HTML page code to initiate the download. An alternate embodiment is a conferencing application that directly communicates with the Operating System 203, and has been compiled into the instruction set of the specific Internet device’s processor.

Once the conferencing application 206a is downloaded and launched, the participant 108 is able to view the user interface and functionality of the conferencing application 206a. The conferencing application 206a may communicate with the web server 102, the database server 110, and the video server 103. In a preferred embodiment, the conferencing application 206a receives information from the web server 102 during the launch of the application 206a in the form of query string variables, and once launched, primarily communicates with the video server 103. Additional information about the pending conference or the user may be requested by the conferencing application from the web server 102. This information may be retrieved from database server 110 or some other source, and returned to the conferencing application in various ways known to one skilled in the art, such as XML. In a preferred embodiment, the video server 103 provides the knowledge of the participants in the conference and other important information to the conferencing application 206a. This knowledge is passed to the video server 103 as in a preferred embodiment, and may or may not communicate with the web server 102 or a database server 110. In a preferred embodiment, the function of the web server 102 is to provide an interface to control the launch of and downloading of the conferencing application 206a. In this alternate embodiment, a similar
interface and control is provided within the OS 203 to launch the application 206b. An example of this alternate embodiment is a conferencing application downloaded by the user from the application store, such as the Apple App Store® or Google Marketplace®, and launched by the user via a touchscreen button.

[0162] A database server 110 connection between either the conferencing application 206b or the video server 103 or both may exist in this alternate embodiment. In the preferred embodiment, the purpose of the database server 110 is to maintain and supply information to control access to the video conferencing services, such as limiting the number of participants using the video service at any given time, keeping track of scheduled conferences, and keeping track of valid participants. This function can be embedded in the video server 103, or can be provided as a separate function in a database server 110 as in a preferred embodiment. Another function that requires access to the conference schedule, typically stored in the database server 110, is the function to respond to a request to an extension of time in the conference. In a preferred embodiment, this type of request is granted if conference seats are available according to the schedule, and the requestor’s account allows it. Other ways of determining suitability of extending a conference will be apparent to one skilled in the art.

[0163] The operation of an Internet device, such as Internet device 104-107 will be apparent to those skilled in the art, and will not be described in more detail herein.

FIG. 3 (Software on Web Server)

[0164] FIG. 3 is a block diagram of the software stack on the web server.

[0165] Referring to FIG. 3, Web Server Software Stack 300 shows the configuration of the web server 103 in a preferred embodiment. The stack 300 runs on a provided hardware platform (not shown), and includes the same type of hardware drivers 302 and Operating System 303 as described in FIG. 2. The specific OS 303 and drivers 302 are typically not the same as OS 203 and drivers 202. For example, the Internet device 200 may be using an operating system from Google, Inc. of Mountain View, Calif., and the web server may be using an operating system from Sun Microsystems, Inc. from Santa Clara, Calif. In fact, hardware 201 may use an entirely different processor instruction set than the Internet device 104-107, or the database server 110, or the video server 103. In fact, the purpose of the layered software and the various communication protocols and languages, such as HTML, JavaScript, and Java, for example, is to free implementations of the various components of the video conferencing system 100 from limitations such as requiring a specific processor instruction set or operating system.

[0166] Running on top of the OS 303 is the web server 304. This software receives requests from the Internet connection, such as HTTP requests, FTP requests, and other types of requests, and processes them. The result is delivered back to the requestor via the Internet connection. The specific operation of a web server will be apparent to those skilled in the art, and thus will not be described in more detail herein.

[0167] Resident on the web server 102, as shown in software stack 300 is the web application 305 and the downloadable conferencing application 304. The web application 305 is a collection of web pages, using technology such as Active Server Pages (ASP or ASP.NET) from Microsoft Corporation, utilizing various scripting languages, such as Visual Basic or C#, or PHP, a freeware scripting language. The purpose of such technology is to deliver dynamic web pages, i.e., web pages where content is derived from information, typically from a database or multiple databases and other information sources, such as cookies and session variables. A web application 305 will be apparent to those skilled in the art, and thus will not be described in more detail herein.

[0168] Also resident and available to the web application 305 is the Downloadable Conferencing Application 304. In the preferred embodiment, this is created with tools from Adobe Corporation using their Flash® and/or Flex® technology. Other similar technologies exist, such as those based on Sun Microsystems’s Java, for example. The choice of technology is primarily driven by the likely availability of the appropriate virtual machine browser plug-in on the participant’s Internet device 104-107. The installed base for Flash® technology is well over 90%, thus making it a very good choice for the preferred embodiment.

[0169] The Downloadable Conferencing Application 304 is automatically downloaded by the Internet Browser 204 in the preferred embodiment using techniques described above. Once downloaded to the participant’s Internet device 104-107, it becomes the Downloaded Conferencing Application 206a. This application then receives information from the web server 102 as described above, including values in the query string, cookies, or via a reference to the data stored in the database server 110. These various methods will be apparent to one skilled in the art, and will not be described in detail herein.

FIG. 4 (Software on Video Server)

[0170] FIG. 4 is a block diagram of the software on the video-streaming server.

[0171] Referring to FIG. 4, Streaming Video Server Software Stack 400 shows a similar software stack as shown for the web server 102, and is a block diagram of the software for the video server 103. This software stack 400 also includes a driver layer 402 to interface to the server hardware (not shown), an Operating System 403, and a video streaming application 404. The video streaming application 404 is provided from a supplier such as Adobe Corporation, and provides the capability for hosting multiple video and audio streams coming in and going out to the various participants’ Internet devices 104-107. This video streaming application 404 also provides a compliant scripting engine 405, for the purpose of enabling conferencing applications to be created by programmers with features and capabilities of their own design and choosing. The Compliant Scripting Engine 405 and video streaming application 404 also provide mechanism for the various copies of the downloaded conferencing application 206a or the Conferencing Application 206b to communicate. The operation and function of the video streaming application 404 and the compliant scripting engine 405 will be apparent to one skilled in the art, and will not be described in detail herein.

[0172] Running on the Compliant Scripting Engine 405 is the Video Conferencing Application Script 406. This script, working in conjunction with the Downloaded Conferencing Application 206a or the Conferencing Application 206b provides the unique functions of the present invention, all of which are running on standard systems, such as the participants’ Internet device 104-107 and video server 103. An alternate embodiment is a compiled program as the Video Conferencing Application. Another embodiment is a com-
plied program that includes both the Video Conferencing Application and Video Streaming Application functions combined, running on the Operating System 403.

[0173] In a preferred embodiment, once the Downloaded Conferencing Application 206a or the Conferencing Application 206b is launched, and any required information is provided, either from the web server 102 for application 206a, or in other ways for application 206b, such as the user providing the information, or the information extracted from a database server 110 using the user login information provided by the user, no additional interaction with the web server 102 is required. All interactions during the videoconference may take place within the application 206a or 206b in conjunction with the video server 103. An example of an exception to this may be the request to extend the conference, as described above.

[0174] In alternate embodiments, additional communications may be desired, such as informing the web server 102 or the database server 110, or the database server 110 via the web server 102 or the video server 103 when a user is quitting the application 206a or 206b. This information can also be generated by the video server 103 when the video stream or other control stream, such as the polling stream described below is terminated, and provided to the database server 110 and/or the web server 102 as another alternative embodiment. This additional communication allows for more effective management of the limited resources of the video-streaming server 103. For example, if the video-streaming server only supports 50 video streams (50 participants), when a conference ends, these seats are now available for other participants to use. This information can be utilized by the web server 102 to allow or disallow additional participants’ access to the video conferencing application 206a or 206b, or to allow or disallow participant access to a conference within the conferencing application 206a or 206b.

[0175] An additional communication that may be desired is a termination command or a notice of upcoming termination, sent from the web server 102 or the video server 103 to the conferencing application 206a or 206b. This type of capability allows the conferencing application 100 to terminate conferences that are going over their preset time limit. An alternate embodiment is to provide the conference end time, and have the conference application 206a or 206b show remaining time, and automatically end the conference when the time runs out. Another alternative embodiment would allow a conference to continue if there are no scheduled conferences that require the conference seats. One mechanism for this is to allow a facilitator to request additional time, as described above. Such an extension of time may update the remaining time displayed by the conferencing application 206a or 206b.

[0176] The following sections will describe a preferred embodiment implemented in a user interface in FIGS. 5, 6, 7, 8, and 9. The figures show both the facilitator and non-facilitator user interface view. Additional facilitator-only functions are also provided via drop-down right-click menus, or other menus or controls panels (not shown). These additional controls can be provided in many different ways apparent to one skilled in the art, and thus will only be referenced herein but not displayed in the interface diagrams. The key operation mechanisms of the present invention will be clear from the diagrams showing the universal features of the conferencing system interface for all participants, including facilitators.

FIG. 5 (Facilitator, Speaking)

[0177] FIG. 5 is a facilitator user interface, with a facilitator speaking to the attendees.

[0178] Referring to FIG. 5, the facilitator’s user interface is shown with the facilitator “speaking” or having the floor.” “Speaking,” in the context of the present invention, generally indicates that a participant “has the floor,” and the speaker’s outgoing video stream is being transmitted to a server, and then forwarded to all of the other participants.

[0179] Facilitator Interface Speaking

[0180] The interface 500 includes a header 501 that specifies the conference name, a video display 502 showing the current speaker (in this case, the facilitator), the current speaker’s name 503 in close proximity to the video panel, an action button 504 indicating the primary current action available, a text display area 505 to display text messages from conference participants, a text input area 506 for entry of text messages, text input buttons 507 for controlling text message submission, an optional status panel 508 showing a color-coded or iconic indication of the state of the interface. One of ordinary skill in the art will understand there are other ways to convey the status information, including in the list of participants. In this example, the status panel 508 is red, with the text “Warning: Camera Live!” This reminds the facilitator that the video camera and microphone are live and transmitting video and audio from the facilitator to the conference attendees. The outgoing facilitator video is being “self” displayed in video display 502, and is being sent to the server to be forwarded to the other participants for display on all of the other participant interfaces in the current group or subgroup. The “self” display (of a speaker’s video display 502 while the speaker is speaking) allows the speaker to adjust his/her position, expression, and lighting as required for best presentation.

[0181] The interface 500 also includes a participant panel 510 showing a list of the current conference participants, along participant type icons and participant status icons 509.

[0182] Specifically, the participant panel shows the following (from left to right) for each participant: a type icon, a status icon, and a name of the participant. Any other relevant information desired may also be displayed for each participant, such as an image or avatar of each participant. The participants may be ordered from top to bottom based on facilitators on top, or based on time of admission into the conference, in alphabetic order, or randomly or by some other logic.

[0183] Type Icons

[0184] In a preferred embodiment, type refers to the type of connection from the participant to the server: a) video (including audio and text, also known as full video, indicated by video camera icon), audio (including text, indicated by a music note and speaker icon), or only text (indicated by an icon with the letter "a"). A special type indicator for a facilitator is shown (509) as a star. In a preferred embodiment, facilitators are required to have a full video connection, and thus the type icon for a facilitator may merely indicate a facilitator type, which is inherently also a video type in a preferred embodiment.

[0185] In a preferred embodiment, it is assumed (default) that the participant can receive video and audio (often called “full video”), can display the video, and can play the audio over earphones or speakers. Thus, the connection type only indicates how the participant can ask questions or make comments: by full video, by audio, or by text (only). It is assumed (default) that all participants can ask questions via text entry,
unless the facilitator has deactivated that feature for the purpose of eliminating distracting chatter in large groups.

[0186] In the case where video cannot be displayed by a participant’s Internet device, due to lack of processing resources (including lack of a display screen) or bandwidth, only the audio will be presented. This preferred embodiment does not support participants that do not have the ability to at least play audio. A common audio-only device is a telephone.

[0187] Status Icons (Color)

[0188] The status icons in the participant list 510 display one of the several possible statuses for each attendee, such as: present, hand raised, text message pending, speaking, or other statuses described below. The status icons in FIGS. 5, 6, 7, 8, and 9 use color to display status. However, a combination of color and graphic elements, or graphic elements alone could be used as well. See FIG. 13 below for additional possibilities.

[0189] In an alternate embodiment, a single combined icon (not shown) will show both the type of attendee (text, audio, video, or facilitator) as well as the status (present, hand raised, text message pending, speaking, etc.). For example, type may be indicated by the shape of the combined icon, and the status may be indicated by the color of the combined icon.

[0190] In a preferred embodiment, status for participants in the attendee list 510 is indicated by color: gray for present, yellow for hand raised (requesting the floor or responding to a poll or question), green for text message pending, and red for currently speaking. Other color assignments or use of graphical elements with or without color-coding will be apparent to one skilled in the art. The current speaker and facilitators may browse text messages by a control, such as a cursor hover over the selected attendee, or by clicking or touching a message pending icon associated with the attendee list element. This allows the current speaker and facilitators to request questions or comments, and, without giving the floor to other attendees, respond to the question or comment, as desired, as described below. Many other possible arrangements will be apparent to one skilled in the art. In an alternate embodiment, message browsing is available only to facilitators. In an alternate embodiment, all text messages are displayed in the Text Display Area 505 only, as soon as they are sent.

[0191] In a preferred embodiment, pending messages are only possible in Closed Mode (a facilitated group). Also in a preferred embodiment, in Group Mode (non-facilitated groups) and in Multi-Interactive Mode (subgroups) all text messages are automatically displayed to the current group or subgroup in a manner similar to a text chat. The term “pending messages” refers to messages that are received by the facilitator but not displayed in the Text Display Area 505, but only become visible to the facilitator in a pop-up area or other display via a control, such as hovering the cursor over an attendee’s name in the list, or touching or clicking a message pending icon that appears next to the attendee name after they send a message. Such a pending message may be released to the Text Display Area 505 by another control, such as clicking or touching the pop-up message display. This technique allows the facilitator to receive questions and comments, but only release for general view those of interest. This type of embodiment is especially useful in large conferences, where free-style messages posted directly into the Text Display Area 505 by the many participants is distracting. Additionally, the function of reviewing and selecting comments and questions is best handled by a second facilitator, allowing the primary facilitator to focus on answering questions or responding to comments.

[0192] In a non-preferred embodiment, text messages are simply automatically scrolled into the text display area for all to see. Text messages may travel independently of video streams, and thus require little overhead. However, this automatic scrolling tends to encourage “side discussions” among participants, and tends to detract from the conference (particularly with large groups).

[0193] Another embodiment is to generate a short sound whenever a message is received and displayed in the Text Display Area 505. This sound informs the participant that a new message has arrived, allowing them to focus on the speaker or on the video camera if they are the speaker. Preferably, this sound can be enabled or disabled by the user. This technique of using specialized sounds to indicate status changes can be applied to other aspects of the interface, such as a sound when receiving the floor, a sound when hands are raised or lowered, or a sound when a new participant signs into or leaves the conference.

[0194] The Text Display Area 505 may become difficult to use if many text messages are displayed. One embodiment is to provide a way to expand the Text Display Area 505 using a control, such as a clickable or touchable icon. The Text Display Area 505 could then be expanded, or displayed in a separate, larger window, or by expanding the size of the current window.

[0195] In many cases, attendees will want to send information out to the group, such as Internet URL’s, names, addresses, or other text. This indicates that the participant who wants to use the information after it has scrolled out of the Text Display Area 505 must scroll up to find what they want. However, if the Text Area Display 505 automatically scrolls to the bottom of the list to show the new message every time a new message is received, the participant will become frustrated and not able to locate and copy the information desired. One embodiment to prevent this from happening only scrolls to the bottom of the messages in the Text Area Display 505 when a new message is received if and only if the Text Area Display 505 is essentially already displaying the last message at the time the new message is received. This can be determined by the scroll position at the time a new message is received. This technique allows the participant to scroll up the list of messages with impunity. The participant can scroll to the bottom of the list of messages to re-enable the normal action when new messages are received.

[0196] Colored Status Icons (Interaction)

[0197] The participant panel 510 in FIG. 5 shows: a red status icon (very dark grey in the black and white figure) for the facilitator Hilda Spanner (indicating a status of leaving the floor); a green status icon (medium grey in the black and white figure) for Tom (indicating a status of having a text message/comment waiting); and a yellow icon (light grey in the black and white figure) for Lisa (indicating a status of waiting to ask a verbal question or make a comment via video). All other icons are dark grey, indicating presence of the participant, but without a request for interaction.

[0198] Message Browsing

[0199] A facilitator or current speaker can view the status in the participant panel 510, and use a mouse or keyboard shortcuts or other controls such as a touch screen to respond to text messages or to grant the floor to another participant. Scanning and viewing text messages is referred to as “Message Brows-
ing,” and is accomplished by a controls as described above. The facilitator or current speaker will see a popup containing the text message sent by that participant. In this case, clicking, double tapping, right-clicking, or other control apparent to one skilled in the art on the participant icon, a message pending icon, or attendee list element will transfer the text message to the text area 505 on all other participant interfaces in the current group or subgroup, tagged with the name of the person who submitted the message. Once a text message has been displayed on all text display areas for the current group or subgroup, the speaker typically reads the text message aloud, or may only refer to it, or give a brief statement summarizing the message. Also, the “pending message” status for the attendee will be deactivated.

[0200] Static, Dynamic, or Streaming Text Message

[0201] The displayed text message is either static, i.e., not changeable by the selected participant or is dynamic. With a dynamic text message, additional typing by the selected participant will show up automatically.

[0202] In a preferred embodiment, the text is static. In this case, the status of the participant in the participant panel 510 will automatically reset to the default “present” gray state once the text message is released and displayed (along with the attendees name or screen name).

[0203] The dynamic text alternative case requires an additional state for the status icon, such as blue, indicating a text stream is active from the selected participant. This state then must be cleared by an additional control by the speaker or facilitator when the interaction is completed, such as clicking or tapping the participants’ icon a second time.

[0204] Releasing the Floor

[0205] If a participant has raised their hand (yellow status), the facilitator can choose to select the participant by clicking, tapping, or other control. In this case, the video server 103 is directed to take the video/audio stream from the selected participant, or audio stream for an audio only connection, and redirect it to all participant interfaces, including the facilitator interface, as described below in reference to FIG. 6. This action is referred to as releasing the floor to another participant. A facilitator always retains control, however. At any point, a facilitator can return the floor to themselves, either by clicking or otherwise selecting (actually “deselecting”) the participant that has the floor, or by selecting their own icon, or pressing a keyboard shortcut or other control.

[0206] The facilitator or participant currently having the floor of course can always release the floor no matter if another participant has their hand up or not. One of the functions of the system of the present invention is to automatically transfer the floor to another participant that has a raised hand when the floor is released by the current speaker, assuming no other action by the speaker or a facilitator has taken place, such as the current speaker selecting the next speaker, or the facilitator transferring the floor to themselves or another speaker. This is called Manual Conversation Flow Control. For the multiple raised hands case, the system automatically selects the first person that raised their hand. This action will take place automatically in a preferred embodiment, and is called Automatic Conversation Flow Control. In another embodiment, this feature may be turned off by the facilitator.

[0207] Any participant can locate their information in the participant panel 510 by looking for their name, or by some other graphical indication. In FIGS. 5, 6, 7, 8, and 9, the name of the participant that is currently speaking is shown in bold. Other graphical identifications will be apparent to one skilled in the art. An alternate method is described below in conjunction with FIG. 13.

[0208] Voting

[0209] In one scenario, a vote or poll is requested by the current speaker, and the voting is performed by a raising of hands (even if one or more participants have text messages pending). In a preferred embodiment, the raised hand state may take precedence over the text message pending. However, once the hands are lowered, either by facilitator or attendee action, the pending message status resumes. Alternatively, a distinct action button may be presented for raising hands (voting). The results of the poll or vote can be displayed within the graphical user interface, as one skilled in the art will appreciate. An example of this is described in association with FIG. 13 below.

[0210] Thus, the embodiment of the described facilitator interface 500 allows a facilitator to: present information (speak), or browse text messages and respond as desired, or allow a participant to speak for interactive discussion (including video questions and comments from all participants). This is all accomplished without echoes or start-stop interactions, thus allowing smooth and rapid group interactions. Finally, in a preferred embodiment, the facilitator has the ability to present slides to the attendees. In this case, the slide panel will appear in the attendees interface (not shown), or in a separate window, and may also appear in the video display 502 of the facilitator.

[0211] In one embodiment, the presentation is launched by a facilitator, using a control (not shown). The presentation slide will then automatically appear in the designated area, or a graphic icon will be presented in the interface to notify the participants that a slide presentation is available. When a participant clicks or taps the icon, the slides will appear. In the case of a small screen device, such as a smart phone, this may activate an alternative panel that the participant can access via touch or gesture to view the current slide. Likewise, the participant may access the video panel via touch or gesture. Other aspects of the operation of this feature will be apparent to one skilled in the art, and will not be described in any further detail herein. In other embodiments, slides may be sent as a video stream, or may be sent as documents (for reduced overhead).

[0212] Simple Control (“Dominant” Action Button)

[0213] A very important feature of the preferred embodiment is the simplicity of control. In FIG. 5, the Action Button 504 indicates a single action that a participant (the facilitator in this case) can take—that is, to release the floor (stop speaking, and perhaps automatically transfer the floor to another participant). This “dominant” action button always has a single function (which may vary dependent upon the context), as will be apparent in the discussion below, and optionally has an easy keyboard shortcut. This action button is described as “dominant” because it may be used to perform the majority of interactions during a conference. This action button may also be described as a primary action button, or key action button.

[0214] In a preferred embodiment, the keyboard shortcut for the “dominant” action button is the spacebar for keyboard-based Internet devices. This is desirable because using a pointing device and graphically selecting the action button requires additional actions, such as grabbing the mouse, moving the cursor over the action button 504, and clicking the mouse. These additional steps take time and concentration.
The specific action will depend on the implementation and control for any given Internet device. For example, a touch screen graphical user interface requires physically touching the action button (a more direct action by the participant than maneuvering a mouse, but still requiring focus and careful hand control, thus taking more time than pressing a fixed-location shortcut button such as a space bar on a keyboard, or a function button on a touch-screen device, or a single key, such as the * key on a telephone).

[0215] The single action functions preferably available for all participants using the action button or related shortcut are as follows:

<table>
<thead>
<tr>
<th>Attender Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listening (someone else speaking)</td>
<td>Raise Hand</td>
</tr>
<tr>
<td>Listening (no one speaking)</td>
<td>Take Floor(^a)</td>
</tr>
<tr>
<td>Hand Raised(^b)</td>
<td>Lower Hand</td>
</tr>
<tr>
<td>Message Pending(^c)</td>
<td>Raise Hand(^d)</td>
</tr>
<tr>
<td>Speaking</td>
<td>Release Floor</td>
</tr>
</tbody>
</table>

\(^a\)For a facilitator, the action is Take Floor in the preferred embodiment
\(^b\)Not available for a text-only attendee; the action in that case is Raise Hand
\(^c\)Not available for a facilitator in the preferred embodiment

[0216] By using the spacebar for keyboard-based Internet devices such as desktop and laptop computers, the attendee or facilitator can rapidly take the specified action. Virtually all of a videoconference interaction process can be controlled by this single button (or its keyboard shortcut). In a preferred embodiment, the action button is a graphical button ("soft key") that changes the graphic depending on the available action in order to make it very obvious what current action is available. In FIGS. 5, 6, 7, 8, and 9, the action button is shown as a standard text button that only changes its text value to reflect the available action.

[0217] Text Entry

[0218] The text entry system for the present invention will now be described, with support from FIG. 5, FIG. 8, and FIG. 9. In the preferred embodiment, text messages from a facilitator are not set as pending, but rather are always sent as described in FIG. 12 below. Briefly skipping ahead, FIG. 8 is provided as a view from an attendee, Tom Wittier, who has posted a text message. The interface can be seen as Tom’s view because his status icon 809 includes bold text of Tom’s name next to it. In an alternate embodiment, the facilitator’s text messages can also be set to pending when someone else has the floor.

[0219] Space Bar (Function Depends on Context)

[0220] The keyboard shortcut that is easiest and fastest to access for keyboard-based Internet devices—the space bar—is also required for text input. Preferably, the text input system is configured to allow the space bar to be used for video conference operation (as a shortcut for the dominant action key) as well as for text input (depending upon the circumstances).

[0221] Text Entry Mode (Enter/Exit)

[0222] Specifically, the text input area 506 is inactive unless activated by the attendee. Activation is by clicking on the text input area 506 to get a blinking cursor, or by pressing the TAB key to get the blinking cursor. When activated, the text input area 506 is highlighted graphically, to reinforce the state change for the user, such as displaying a color background in the text entry area, or drawing a colored line around the text area.

[0223] When the text input area 506 is active, the space bar is strictly a text space bar, and is not available for interaction with the “dominant” action button 504. Conversely, when the text input area is inactive, then the space bar is strictly available for interaction with the “dominant” action button, but is not available as a text space bar. This detailed logic makes keyboard interaction much more user friendly.

[0224] Once the text input area 506 active, the attendee can type a message, and use standard editing keys, such as backspace, arrow keys, copy, paste, undo, etc. to create a text message. The space bar is used to generate spaces in the text. Preferably, when there is an incomplete message (some text entered, but not yet sent) in the text input area 506, both the Send button and Clear button 507 will be active (bold, black text rather than dimmed text). The state of a button may also be indicated by highlighting, changing color, or changing the graphical appearance of the button.

[0225] The attendee may exit the text entry mode with an incomplete text message at any time by pressing the TAB key, or by clicking on the action button 504. This feature is not shown in Table 1. The text input area highlight and blinking cursor will disappear, reinforcing the mode change for the user. The text input Clear button 507 will retain its existing state (dimmed or not dimmed) for both active and inactive state for the text input area 506, thus allowing an already entered or pending message to be cleared even if the text input area 506 is not active. In an alternate embodiment, the Send button (and its related keyboard shortcuts) also retains its existing state, allowing a text message that has not been sent to be sent without reactivating the text entry area 506.

[0226] The attendee may return to the text entry mode by pressing the TAB key, or clicking on the text entry area, or by other controls.

[0227] The attendee may exit the text entry mode and simultaneously send the message to the server at any time by pressing the Return or Enter key, or clicking on the Send button 507. In this case, the Send button 807 will go dim (inactive), as shown in FIG. 8.

[0228] If the system is in group or subgroup mode, the sent text message will be displayed with the attendee name in the text display area 905 for all group or subgroup members, as shown in FIG. 9. In facilitated (Class) mode, the message will be pending (pending selection by the facilitator), as shown in FIG. 8, and will change the status of the sender to the message pending status, as shown by status icon 809. In this case, the text will remain in the text entry area 506 (preferably dimmed) until transferred to the text display areas of all attendees by facilitator action (as shown in FIG. 9, text display area 905), or until the attendee clicks on the active Clear button 807. The Clear button 807 will be active until the message is cleared or transferred. The ESC (Escape) key is a keyboard shortcut for the Clear button in one preferred embodiment.

[0229] As described above, the attendee can exit the text entry mode and simultaneously delete any incomplete text message at any time by clicking on the Clear button 507 or by
pressing the keyboard shortcut ESC key. In this case, both text input buttons 507 will be inactive (dimmed), and any text in the text entry area 505 is cleared. The attendee can clear a pending message at any time by clicking on the Clear button or pressing the ESC key. The text input mode does not have to be active for this action to take place.

[0230] An alternate embodiment removes both the Send and Clear buttons. In this case, a message can only be sent using the Return key, and clearing a message can only be accomplished by selecting it and using the cut or delete keyboard functions. Finally, another alternate embodiment, either with or without the Send and Clear buttons, includes only a single mode for text entry and display is supported (typically the group mode). Optionally, a setting, available to a facilitator, could be provided to switch between the alternative embodiments.

[0231] The value of the facilitated text entry mode is especially valuable when working with large groups, where one facilitator manages the messages and the other manages the conversation flow.

FIG. 6 (Facilitator, Listening)

[0232] FIG. 6 is the facilitator user interface, with the facilitator listening to an attendee.

[0233] Referring to FIG. 6, the facilitator's user interface (with the facilitator listening) shows the interface response to the facilitator having selected Lisa Jones (from the participant panel 506) by using a control such as clicking, tapping, or by other controls apparent to one skilled in the art. The interface displays the selected attendee's video display 602 and speaker name 603. The status panel 608 is grey and displays "Listening..." as the current status for the facilitator. The selected speaker, Lisa Jones, is indicated by the red color of the status icon 609 of Lisa Jones. The facilitator's status icon is grey (present, not speaking), and the action button 604 for the facilitator is labeled Take Floor. The facilitator has additional actions not available to a non-facilitator. For example, the facilitator may click on his own icon to take back the floor, or the facilitator may use the default or "dominant" action indicated by action button 604. In an alternate embodiment, the action button operates the same for a facilitator as for a non-facilitator. In that embodiment, the action button would display "Raise Hand" to the facilitator.

[0234] This additional facilitator functionality may be provided in a preferred embodiment by providing a separate downloaded conference application 206a for a facilitator. An alternate embodiment includes all functionality within a single downloaded application 206a where the additional facilitator functionality is enabled by status information transferred by the web application 305 to the downloaded application 206a as described above. An alternate embodiment is to provide a graphic mechanism, such as a Settings button, for example, that allows a participant to view a settings panel. The button may only display for facilitators, or alternatively, the settings panel may show limited options for non-facilitators, and more options for facilitators. In yet another embodiment, all options are shown to all participants in the settings panel, but options not available to a non-facilitator are dimmed out. This approach has the benefit of making all attendees aware of the available functionality of the system, thus potentially attracting them to facilitate their own meetings.

FIG. 7 (Attendee, Hand Raised)

[0235] FIG. 7 is an attendee user interface, with a hand raised.

[0236] Referring to FIG. 7, the attendee interface with hand raised shows the interface response to the attendee Lisa Jones having selected the default action "Raise Hand," either by clicking on the action button 704 or pressing the keyboard shortcut. Note that the Lisa Jones name is in bold in the attendee list, indicating we are viewing her view of the application interface, and that Hilda Spanner is currently the speaker. The status panel 708 and the Lisa Jones participant status icon 709 have changed color in response to the raise hand action, indicating that the attendee's hand is raised. The text in the status panel 708 also states "Hand Raised!" Finally, the action button 704 now indicates the primary or dominant action for the attendee: "Lower Hand." All attendees that have their hands raised will see a similar status on their interfaces, and can use the action button or keyboard shortcut to lower their hand, or click on their status icon to lower their hand. The current speaker also has the ability to click on another participant status icon, which transfers the floor to that participant (hand raised or not raised). In a preferred embodiment, transfer of floor from one participant to another also clears all raised hands automatically. Alternatively, the current speaker may only transfer the floor to another participant who has their hand raised.

FIG. 8 (Attendee, Message Pending)

[0237] FIG. 8 is an attendee user interface, with a message pending.

[0238] Referring to FIG. 8, attendee interface with message pending, the interface response to an attendee sending a message when not having the floor is shown via the status panel 808 and status icon 809. In this example, this is the interface viewed by Tom Wittier, as indicated by the bold text for Tom's name in the attendee list. In a preferred embodiment, messages from a facilitator or speaker are always transferred to the text display area for all subgroup or group members. In Class Mode, where all attendees are in the same group with a facilitator, all sent messages from attendees that are not the current speaker and not a facilitator are set as pending, with a status icon indication (change in color or the appearance of a Message Pending icon) and a status panel indication ("Message Pending...") as shown in the user interface of the attendee. In an alternate embodiment, when a pending message is accepted and displayed, all other pending messages are retracted, and returned to the Text Entry Area 806 of each attendee, respectively. This allows each attendee that has sent a message to re-evaluate and either clear the message or resend it.

[0239] When subgroups are active, and a facilitator in the main conference room, facilitator text messages are delivered to all participants along with an alert sound. This function allows the facilitator to notify the subgroups that time is running out, or to deliver additional instructions. In an alternate embodiment, received text messages are always delivered with an alert sound, unless the attendee has deactivated that feature using a settings panel or other way.

FIG. 9 (Attendee, Message Displayed)

[0240] FIG. 9 is an attendee user interface, with a message displayed.

[0241] Referring to FIG. 9, the attendee interface with message displayed shows the interface response to the current
speaker (Lisa Jones) having clicked on the green message-pending status icon of Tom, thus releasing his message to all participants. The message is displayed in text area 905 for all participants, with the name of the person providing the text message (Tom Wittier in this case), and the action button 904, status panel 908, and status icon 909 revert to the default state for listening. In this case, the interface will display the message and remove the pending state. Lisa would now respond to the comment/question, and optionally give the floor to Tom. In an alternate embodiment, only a facilitator can review and accept pending messages.

Class Mode (At Least One Facilitator)

The operation of the present invention has been described above in relation to a videoconference between one or more facilitators and a group of attendees (in a first mode referred to as Class Mode). In addition to Class Mode, two additional modes are supported by the present invention.

Group Mode (No Facilitator)

The second mode, referred to as the Group Mode, is supported when there is no designated facilitator. This group mode is preferred when a participant schedules a meeting with other equal participants (rather than a facilitator scheduling a meeting with attendees). This group mode is also used in conjunction with the third mode.

Multi-Interactive Mode (Subgroup Mode)

The third mode, referred to as the Multi-Interactive Mode (or Subgroup Mode), supports multiple simultaneous subgroups with a facilitator for the overall group. For this subgroup mode, each subgroup preferably operates as a relatively independent small group (in group mode), with the facilitator of the overall group also overseeing the subgroups. These additional modes will be described below. Alternatively, the subgroups may operate in Class mode, with each subgroup having a subgroup facilitator.

In the Group Mode, there is no facilitator. However, there is preferably a “conference owner” (the participant that scheduled the conference). In one embodiment, the conferencing system may give the same additional features and functions normally given to a facilitator to the conference owner, essentially making them identical to a facilitator. The conferencing system 100 preferably automatically gives the floor to the first participant at the beginning of the conference, as described below in conjunction with FIG. 12. Alternatively, the floor is open at the beginning of the conference. As described above, the interface for non-facilitators is illustrated by the interface shown in FIGS. 7, 8 and 9, and facilitator’s interface, shown in FIGS. 5 and 6, may have additional functions (not shown). Also see the description below in conjunction with FIG. 13 for a different embodiment of an interface.

FIG. 10 (Multiple Conference Modes)

FIG. 10 illustrates multiple conference modes.

Referring to FIG. 10, the Class Mode 1001 (top left portion of the figure) shows multiple participants comprising: a facilitator 1010 and some other participants (attendees) 1011. The Group Mode 1002 (top middle portion of the figure) shows only participants 1011, with no facilitator. Finally, the Multi-Interactive Mode 1003 (subgroup mode, bottom left portion of the FIG. 10) shows three subgroups (each with three participants), a facilitator 1010, and one or more optional subgroup facilitator slot 1012 for each of the subgroups.

While FIG. 10 shows three subgroups (each subgroup having three participants) in the Multi-Interactive Mode 1003, the number of participants in each subgroup can be set to any number from two to ten (or more) by the facilitator, and thus the number of actual subgroups formed will vary, depending on how many participants are in the overall group. For example, if there are 24 participants, and the facilitator requests subgroups of size 4 (four participants maximum in each subgroup), then six subgroups of size 4 will be formed, as described below.

Referring back to FIG. 5, when in Group Mode in a preferred embodiment, the text interface (text display area 505, text entry area 506, and text input buttons 507) operates much like a standard chat room, i.e., anything typed and sent by the control (in FIG. 5 sent via a Send button click or by pressing the return or enter key) will show up (along with the participant’s name or identifier) in the text display area 505 (as shown in FIG. 9, text display area 905). Once typed and sent, it cannot be recalled, and all participant text messages are displayed in real time. Like recalled email, it is possible to implement procedures to recall comments, but these recall procedures are not worth the overhead.

One problem that can occur in Group mode is when a participant wants to talk for a long period, and not give the floor to other participants. There are a number of solutions for this. In a preferred embodiment, the text message function will provide a way for other attendees to remind the speaker to give others a chance to speak. Other embodiments will be apparent to one skilled in the art. For example, a majority of the participants may be able to steal the floor, or the conference owner may have the ability to take back the floor.

The Multi-Interactive Mode of the present invention is selected by a facilitator by using a provided control. For example, a button, or a drop-down menu from a right-click, a settings panel, or other control may be provided to the facilitator. The control device may provide a selection of previously defined static groups, along with the dynamic group option. The dynamic group option allows a facilitator to specify the subgroup ideal size, such as 2 per group, 4 per group, etc. Of course there may be additional participants that do not form a group of the desired size. For example, if there are 18 participants and a facilitator requests groups of 4, four groups of 4 will be generated, and there will be two participants left over.

Leaver Participants (Three Possible Preferences)

A preference setting may control how these “left over” participants are handled. In a preferred embodiment, the default preference is to distribute extra participants among the groups. Thus, two groups of 5 and two groups of 4 will be created in the example cited. This is referred to as the group size plus (+) preference. An alternative preference is to reduce the size of groups by one to build up the smaller leftover group. This is referred to as the group size minus (−) preference. In this example, one participant will be taken from a group of 4 in order to fill out the last group, resulting in 3 groups of 4 and 2 groups of 3. Finally, a third alternative is to keep the leftover participants together as a smaller group. This is referred to as the default group size preference.

Group Size Plus Preference (+)

Returning to FIG. 10, the Multi-Interactive Participant List 1004 (bottom right portion of FIG. 10) shows how the class will be split up into subgroups when the facilitator has selected the “+2” group size preference. The preference is displayed by putting a + or − sign after the selected group size, or showing no sign after the selected group size, as can be
seen in the header of the Multi-Interactive Participant List 1004. Also displayed in the header is the operating mode for the conference, shown here as Class Mode. This indicates that groups have been formed, but the entire set of participants and the facilitator are still together in the Class Mode. In this mode (actually a subgroup setup aspect of Class Mode), facilitators have the option of dragging and dropping participants to change the group layout, as desired. Other controls apparent to one skilled in the art can be utilized for editing group arrangements. In an alternate embodiment, the facilitator may also change their mind and end the setup mode, or choose a pre-defined static group.

[0259] Subgroup Breakout (Optional Facilitator Visits)

[0260] Once a facilitator has instructed the class regarding what exercise or discussion is to take place within the subgroups, the mode can be switched to Multi-Interactive Mode using a control, such as clicking on the Participant list header 1004, or right-clicking and selecting an appropriate command from a drop-down menu, or selecting a command from a settings panel. Once this action has been taken (subgroup breakout), each subgroup is on its own, and is working within the Group Mode for its own subgroup. At that point, the subgroup members will only see their co-members in their participant list, or other participants in other groups will be displayed in a dimmed manner. Facilitators continue to see all groups in their participant list. The facilitator may at this point name this grouping and add it to the list of defined static groups for possible use later.

[0261] When the subgroup breakout function is initiated, the facilitator remains in the main conference room, and thus cannot hear or see anyone in the video display area (the “no speaker” or “Floor Open” notification video or graphic is displayed, since no one is currently speaking in the main conference room). This is akin to having a class break up into groups, each group going into a separate room nearby and closing the separate room’s door, while the facilitator remains in the main auditorium. However, as shown in FIG. 10, Facilitator Slots 1012 exist within each subgroup. These slots simply indicate that a facilitator (or, for that matter, multiple facilitators) may temporarily (or permanently) join the subgroup. The slots are not currently populated, but are used here to show possible locations for the facilitator other than in the main conference room. Thus, a facilitator can enter any of the subgroup rooms and hear what is being said simply by clicking on an appropriate unpopulated facilitator slot, or clicking on the selected group header bar above the group in the Multi-Interactive Participant list 1004, or by some other control apparent to one skilled in the art. When entering a subgroup, the facilitator’s icon appears in all subgroup participants’ attendee lists (and then disappears when the facilitator leaves).

[0262] Once in a subgroup, a facilitator has the ability to take control, or remain passive. When the facilitator is present but remains passive, each participant continues to operate in effectively a Group Mode. The subgroup members will see the facilitator appear in their participant list, and thus will be aware that the facilitator is listening. However, the facilitator may take control of the group by clicking on the floor, or by being selected to speak by a subgroup member. When the facilitator leaves the floor, or transfers the floor to another participant, control returns to the group. Finally, a facilitator may leave the room and return to the main conference room by clicking on the facilitator icon 1001, or clicking on the Multi-Interactive Participant List 1004 header, or other control. Alternately, the facilitator may leave a first subgroup and transfer directly to a second subgroup by clicking on an unpopulated facilitator slot of the second subgroup, or by clicking on the selected group header bar, or by other control.

[0263] Thus, while the subgroups are meeting during a breakout, a facilitator has the ability to “walk around” and listen in to subgroup activities, and enter the conversation, if desired. Alternatively, subgroup facilitators may be assigned or elected.

[0264] In one preferred embodiment, when a facilitator is in the main conference room, any message typed and sent by the facilitator will appear in all participants’ text area 505. This allows a facilitator to give notice that the participants should finish up, or give them an update on how much time they have left. Also in a preferred embodiment, any such broadcast facilitator text message to subgroups will be accompanied by an alert sound (such as a bell). For embodiments that include a sound alert for all received messages, the alert sound from the facilitator may be a different sound to distinguish a message of higher importance has been received.

[0265] Any message typed by a facilitator when visiting a subgroup (while populating a subgroup facilitator slot) will only be seen in the subgroup text display area. In an alternate embodiment, a facilitator typing a message even in a subgroup will be seen by all participants. In an alternate embodiment, the overall group facilitator may participate as a mere participant in a subgroup, perhaps to fill in a subgroup up to the desired number of participants.

[0266] Reconvene Conference

[0267] A facilitator can return the conference to the Class Mode (reconvene conference) by a control, such as selecting a command from a drop-down menu or panel or other controls apparent to one skilled in the art. In this case, all groups will be returned to the conference room, but (optionally) will still be displayed in a grouped manner in the attendee lists, as shown at the bottom right of FIG. 10. Thus, a facilitator can issue a new breakout command to have the same subgroups meet again for additional work, or can command the system to create new subgroups, or can select from a set of pre-defined static group definitions. This is one mechanism for operating with dynamic groups rather than static groups. By not creating a new group arrangement and using the existing one, the facilitator is using a static grouping. However, if the facilitator wishes to use a different group arrangement, and then return to the current one, the current arrangement should be saved by name into the pre-defined static group list.

[0268] An additional control for subgroup participants may be desired to enable them to indicate that they are done with whatever work was requested for the subgroup by the facilitator. This additional button is referred to as the Exit Subgroup button. This additional button appears in the interface of subgroup participants when in a subgroup during a Multi-Interactive Mode. An additional status icon status preferably also becomes available, so that all subgroup participants can see which participants are finished (have clicked on the Exit Subgroup button).

[0269] The exit function is preferably a toggle, and thus can be turned off if the participant changes their mind, and is also turned on automatically by any other action in the interface by the participant. When all subgroup members are finished (i.e., all have set the exit status), the subgroup is disbanded (effectively by unanimous decision), and subgroup members are either returned to the main conference room, or are reassigned by the system to another subgroup, such as in the case
of a mill interaction. This additional exit action control can be provided in various ways other than a button, as will be apparent to one skilled in the art.

[0270] A preferred embodiment thus supports: a) a Class Mode similar to a classroom, with text message browsing, responding to raised hands, and allowing discussion under the control of one or more facilitators; b) a Group Mode, where a group of participants can interact on a peer-to-peer basis; and c) a Multi-Interactive Mode, where a conference is broken up into small subgroups, where facilitators can roam, listen in and participate with any subgroup, and communicate to the entire group via text messages.

[0271] An alternate embodiment supports a Directed Meeting mode similar to Group Mode, with a facilitator that has full control. Specifically, the text entry function works like the Group Mode with a facilitator in the meeting. This embodiment is described in conjunction with FIG. 13. Various combinations of these modes are possible in other embodiments, as will be apparent to one skilled in the art.

FIG. 11 (Action Commands)

[0272] FIG. 11 illustrates multiple action commands.

[0273] Referring to FIG. 11, a command table illustrates an exemplary set of commands (or requests) available to various types of parties, the types being indicated by columns. An “X” indicates the command is allowed. Notes 1) through 6) at the bottom of the figure indicate additional information regarding certain allowances.

[0274] The illustrated parties are (columns from left to right): Facilitator VAI (with Video, Audio, and Text); Attendee VAT (with Video, Audio, and Text); Attendee A1 (with Audio and Text only); and Attendee T (with Text only).

[0275] Other possible parties are not illustrated. For example: Attendee A (with Audio only); Attendee V (with Video only); and Attendee VT (with Video and Text only). Under some circumstances, there may be more than one Facilitator.

[0276] For example, the “Extension” command is only allowed to the Facilitator. The “Subgroup Done” command is only allowed to the Attendees, and is not allowed to the Facilitator. However, the facilitator has a “Reconvene” command, which is not allowed to the Attendees.

[0277] A lock command prevents new attendees from entering the meeting. Unlock removes that restriction.

FIG. 12a (Conference Start Request)

[0278] FIG. 12a illustrates a first portion of a conference start request diagram.

[0279] Referring to FIG. 12a, a flow diagram for a preferred embodiment, in step 1200 the conference application starts by receiving a request for a video conference in step 1201 along with conference information, such as time and length of conference, and attendee list (the term “attendee” is used broadly in FIGS. 12a to 12c, as referring to participants including facilitators and non-facilitators). Alternatively, the attendee list is dynamically constructed as attendees arrive.

[0280] Validate

[0281] In step 1202 the system validates the request, such as by comparing the request with the conference schedule database, or by checking for available seats. If the request is not valid, the conference application exits in step 1203. Otherwise, in step 1204, the system determines the available data stream from the requestor (text only, text plus audio, or text, audio and video). This information is used to control the attendee type icon. The system also identifies at this point if this attendee is a facilitator, which is also used to control the attendee icon(s), as described previously. If the attendee is a text-only attendee, control is transferred to entry point SL to set the attendee as a listener. In the alternate embodiment described below for step 1205, all new attendees are assigned as listeners.

[0282] Set up as Speaker

[0283] Next, in step 1205, the attendee is set up as the speaker. Step 1205 is also the entry point SS for all cases where an attendee is to be set as the speaker, where the current group is identified, and all current group or subgroup members receive the data stream from the attendee/speaker. All temporary states (such as raised hands) are cleared, and the attendee lists for all attendees are updated with the new status. In an alternate embodiment, the first attendee is always set as a listener (because there is nobody else available to speak to yet). In an alternate embodiment, no speaker is automatically assigned the floor at the beginning of a conference. In this case, no conversation will occur until an attendee takes the floor. In an alternate embodiment, raised hands are not cleared when a new speaker is assigned.

[0284] Attendee Request

[0285] In step 1206, also the entry point L for the Main Loop, the system checks for an attendee request. If one is found, control is transferred to entry point B in FIG. 12b. Otherwise, check for a request to add a new attendee in step 1207. If such a request is pending, go to step 1211 to process the request. Otherwise, check for conference over in step 1208. The conference is over if a timer has timed out, and no conference extension was requested or issued. In this case, exit the conference in step 1209. Next, in step 1210, check for a mill in process. If there is no mill in process, the main loop repeats by returning to entry point L. If a mill is in process, check for available unattached attendees and assign to new subgroups according to the mill process specified above. All new subgroups created start with no speaker assigned. Then the main loop is repeated.

[0286] Validity

[0287] For an add attendee request in step 1211, the request is checked for validity, and the lock status for the conference is also checked. If not valid, or the lock is set, the request is ignored (or expressly denied), and the main loop is repeated. If the request is valid and the conference is open (not locked), then the data stream type for the new attendee is received in step 1212.

[0288] Attendee Type

[0289] Next, the attendee type (facilitator or non-facilitator) is checked in step 1213. If not a facilitator, the new attendee is set as a conference listener in step 1214. Step 1214 is also the entry point SL for setting an attendee as a listener. Step 1214 sets the current data stream from the speaker in the current group or subgroup to be sent to the attendee, and updates the attendee status in attendee lists. If there is no speaker, the “no speaker” notification data stream is sent or the “Floor Open” graphic is displayed instead. Next, the main loop is repeated. If the new attendee is a facilitator, in step 1215 a check is made to see if there is a current speaker. If there is a current speaker, control is transferred to entry point SL (step 1214) to set the new facilitator as a listener. If no speaker is currently assigned, the facilitator is set as the cur-
rent speaker via transfer to entry point SS (step 1205). Alternatively, as described earlier, all new attendees are assigned as listeners.

**FIG. 12b**

[0290] FIG. 12b illustrates a second portion of the conference start request diagram.

[0291] Raise Hand

[0292] Referring to FIG. 12b, entry point B for an attendee request, the possible requests are listed in FIG. 11. In step 1220, the request is checked for Raise Hand Request. In step 1221, a Raise Hand request is processed: the status for the attendee is changed, both in the attendee user interface and all attendee lists where the attendee is displayed. Next, the mail loop is repeated. In one embodiment, if this is the first hand raised (First Hand Up), an alert sound is generated for the speaker. The purpose of this alert sound is to enable the speaker to focus on the video camera rather than having to check to see if anyone has raised their hand. This alert sound should be distinctive, and different from other alert sounds.

[0293] Lower Hand

[0294] In step 1222, the request is checked for Lower Hand Request. In step 1223, a Lower Hand request is processed by updating the status for the attendee user interface and in all attendee lists where the attendee is displayed. If there is a pending text message from this attendee, that state is reasserted. Otherwise, the default present or listening state is asserted. Next, the main loop is repeated. In one embodiment, if this is the last hand lowered (Last Hand Down), an alert sound is generated for the speaker, for the same reason as for the raised hand case. The alert sound for First Hand Up and Last Hand Down should be different, such as a rapid rising or falling tones.

[0295] Give Floor

[0296] In step 1224, the request is checked for a Give Floor (Transfer Floor) request. In step 1225, the Give Floor request is processed by selected attendee to receive the floor is set as the attendee (unless the selected attendee is a text-only attendee), and transferring control to entry point SS to set the selected attendee as the speaker for the current group or subgroup in which the selected attendee is currently seated. If the selected attendee is text-only, the request is ignored, and the main loop is repeated (not shown).

[0297] Take Floor

[0298] In step 1226, the request is checked for a Take Floor Request. In step 1227, the Take Floor Request is processed by first checking the attendee type. If the attendee is a facilitator, control is transferred to entry point SS to set the attendee as the speaker. If the attendee is not a facilitator, in step 1228 the conference application checks to see if there is already a speaker. If not, control is transferred to entry point SS to set the attendee as the speaker for the current group or subgroup. Otherwise, no action is taken, and the main loop is repeated. Alternatively, the Take Floor request when another attendee is speaking is handled as a Raise Hand request. A Take Floor request is ignored if the attendee is a text-only attendee (not shown in the diagram).

[0299] Text Message

[0300] In step 1229, the request is checked for a Text Message Request. In step 1230, the Text Message Request is processed by the conference application by first checking the type of the attendee. If the attendee is a facilitator, in step 1231, the text message is sent to all attendees in the group or subgroup tagged with the sender's name, and an alert is sounded if the message is from the facilitator in the main conference room during Multi-Interactive (subgroup) Mode. As described earlier, in alternate embodiments, messages are always sent to all attendees, and alert sounds are generated unless the attendee has turned them off. Next, the main loop is repeated. If the attendee is not a facilitator, in step 1232, a check is made by the conference application to see if the attendee is currently speaking. If so, control is transferred to step 1231 and the message is sent to all attendees in the current group or subgroup. If the attendee does not have the floor, the conference application checks the current mode. If not in Class Mode (i.e., in Group or Multi-Interactive Mode), the text message is treated as a chat message, and control is transferred to step 1231 and the message is displayed. Otherwise, in step 1234 the Pending Message status is established for the attendee in the attendee interface, and all subgroup or group member attendee lists are updated with the new attendee status. Next, the main loop is repeated.

[0301] Answer

[0302] In step 1235, the request is checked for Answer Request. This request is generated when a speaker clicks on an attendee in the attendee list with a pending message.

[0303] Answer Requests are processed in step 1236, where the pending message selected by the speaker is posted, along with the name of the attendee who sent the message, in all group or subgroup attendees' text display areas 505, and the message pending status for the selected message source attendee is reset or cleared. In one preferred embodiment, the Answer Request can only be sent by a speaker. All attendee lists are updated, and the main loop is repeated. In an alternate embodiment, a facilitator can only send Answer Requests. In another embodiment, when an Answer Request is received, all pending message states are cleared, and the sending attendees request is removed from the Text Input Area of the sender and displayed in all attendees Text Display Areas, and all other pending messages are returned to the respective Text Input Areas to enable the attendees to determine if they wish to respond.

[0304] Extension

[0305] In step 1237, the request is checked for Extension Request. This request can only be generated by a facilitator or conference owner in a preferred embodiment, and represents a request to extend the time for the conference beyond the original set end time. The request may include an amount of time for the extension, or a default amount of extension may be assumed. In step 1238, the conference schedule is checked to see if conference seats are available for an extension. In step 1240, if not enough seats are available, the request is ignored, and the main loop is repeated. Otherwise, in step 1239, the extension is granted, the schedule database is updated, and the conference is extended. If remaining conference time is displayed in attendee interfaces, it is updated.

[0306] Next, the main loop is repeated. In an alternate embodiment, the account status of the conference owner is checked to see if an extension is covered by their account. For example, they may have an unlimited meetings account, which would allow them extensions. Alternatively, one or more seats may have been paid for, and an extension will require additional payment. This can be managed automatically via prepaid seat-hours. In this case, the extension time multiplied by the number of attendees that hold paid seats rather than subscription (unlimited) seats is computed and compared with available seat-hours in the owner's account. If there are insufficient seat-hours, the extension is rejected. Otherwise, the extension
is granted, as described above, and the calculated number of seat-hours is deducted from the owner’s or facilitator’s account.

[0306] Lock

[0307] In step 1241, the request is checked for Lock Request. In step 1242, the request is processed by setting the conference lock on, and the main loop is then repeated. Note that in a preferred embodiment, only a facilitator can lock the conference, preventing new attendees from joining. Typically, for embodiments that support the lock request, an Unlock request is also possible (described below).

[0308] Drop Attendee

[0309] In step 1243, the request is checked for Drop Attendee Request. This request is generated when an attendee Internet device 104-107 disconnects from the video streaming server 103 due to an interruption of the Internet connection 110, due to a failure or crash in the Internet device 104-107, or when the attendee closes the downloaded conference application 206a or conferencing application 206b. In one embodiment, this situation is detected by the loss of a polling signal from the Internet device 104-107. In step 1244, the request is processed first by terminating any data stream to or from the dropped attendee, and removing the attendee from all attendee lists. In step 1245, if there are no more attendees (this was the last to drop out of a conference), then in step 1248 the conference is terminated. If there are still attendees in the conference, in step 1246 the status of the dropped attendee is tested. If the dropped attendee was not speaking, no further action is taken, and the main loop is repeated. If the dropped attendee was the current speaker, in step 1247 the data stream to all remaining members of the group or subgroup of the dropped attendee are sent the “no speaker” notification data stream, or a “Floor Open” graphic is displayed, and the main loop is repeated.

FIG. 12c

[0310] FIG. 12c illustrates a third portion of the conference start request diagram.

[0311] Give up Floor (Release Floor)

[0312] Referring to FIG. 12c, in step 1250, the request is checked for Give Up Floor (Release Floor) Request. In step 1251, the request is processed first by checking the conference mode. If the request to release the floor was given in Class Mode, where a facilitator is present, in step 1252 the facilitator that most recently had the floor is set as the attendee and control is transferred to entry point SS to set the selected facilitator as speaker. In another embodiment, this step is skipped, and processing continues in step 1253. Otherwise, in step 1253, the attendee status for all other attendees in the group or subgroup is checked to see if any attendee has a hand raised. If so, in step 1254, the first person that raised their hand is set as the attendee, and control is transferred to entry point SS to set them as the speaker. This feature requires that the system keep track of the time each attendee raises their hand, so the earliest action can be determined. This automatic transfer based on earliest hand up is called Automatic Conversation Flow Control. Otherwise, in step 1255, the number of attendees in the current group or subgroup is checked. If only 2 attendees are in the current group or subgroup, in step 1257 the other attendee is set as the attendee and control is transferred to entry point SS to set the other attendee as the speaker. If there are more than 2 attendees in the current group or subgroup, in step 1256, the floor is released, the “no speaker” data stream is sent to all attendees in the current group or subgroup or a “Floor Open” graphic is displayed, and the attendee lists and attendee interfaces are updated. The main loop is then repeated. In an alternate embodiment, the two-attendee case is not detected or handled differently than the more than two attendee case.

[0313] Subgroup Exit

[0314] In step 1258, the request is checked for Subgroup Exit Request. This request can be issued by any subgroup member by clicking on the Exit Subgroup button, or by some other control.

[0315] In step 1259, the request is processed first by toggling the exit status for the attendee, and the attendee list status for the attendee is updated. In a preferred embodiment, the icon is gray, with a blue upward pointing arrow to indicate the exit status has been selected. In one preferred embodiment, if the attendee sends any other request after setting the Exit status, the exit status will also be revoked (not shown in the flow diagram). Also in step 1259, the exit status is checked for all other subgroup members. If all subgroup members have not set the exit status, no further action is taken, and the main loop is repeated. If all subgroup members have set the exit status, in step 1260 the conference application checks to see if a mill is currently running. If not, in step 1261, the attendees in the subgroup are released from the subgroup back into the main conference room, and receive the main conference room data stream. If no one is speaking in the main conference room, then the attendees receive the “no speaker” data stream or “Floor Open” graphic. All attendee lists are updated appropriately, and the main loop is repeated. If a mill is running, in step 1262 the subgroup attendees are set into a waiting state, and are made available to the mill processing step 1216 during the next iteration of the main loop.

[0316] In step 1263, the type of attendee making the current request is tested. If not a facilitator, the main loop is repeated. This eliminates testing for requests that cannot be sent by a non-facilitator.

[0317] Create Subgroup

[0318] In step 1264, the request is checked for Create Subgroup Request. This request is processed starting with step 1265, where the conference is locked, preventing additional attendees from joining. In an alternate embodiment, a conference lock is not used, and additional attendees are added to subgroups that have less than the desired number of attendees or start a new group with some existing attendees, based on the Size+, Size- and Default described earlier. In step 1266, the type of subgroup formation is determined. Dynamic subgroup formation is handled by step 1267, where groups are formed according to the settings provided by the facilitator, including group size, group type (mill or standard), and with the preferences for how to deal with excess attendees (Size+, Size- or Default). In step 1269, the newly created subgroups are displayed in the attendee lists, and the main loop is repeated. Static subgroups are handled in step 1268, where the selected group specification is received and implemented. Static group specifications can be provided in a variety of ways apparent to one skilled in the art. Once status groups are formed, in step 1269 the static groups are displayed in the attendee lists, and the main loop is repeated.

[0319] Breakout Groups

[0320] In step 1270, the request is checked for Breakout Groups Request. Subgroup breakout is the event where groups previously formed according to the Create Subgroup Request are instantiated, i.e., each subgroup is set into their one, separate conference room. If no groups have been
defined, the Breakout Group request cannot be sent, or is ignored (not shown). Next, in steps 1271, 1272, and 1273, each of the defined breakout groups are created, one after the other. In step 1271, the request is processed by selecting a first formed subgroup to be broken out. In step 1272, the data streams to the subgroup attendees are set to the “no speaker” data stream. In step 1273, the subgroup members’ attendee lists are updated to show only the current subgroup. The subgroup also disappears from all other group members lists, except for facilitators’ attendee lists. In an alternate embodiment, attendees not within the group are not removed from the list, but are dimmed. In step 1274, the conference application tests to see if all subgroups have been broken out. If not, the next subgroup is selected in step 1271. If all subgroups have been broken out, the main loop is repeated.

FIG. 12d

[0321] FIG. 12d illustrates a fourth portion of the conference start request diagram.
[0322] Join Subgroup
[0323] Referring to FIG. 12d, in step 1280, the request is checked for Join Subgroup Request. This request can only be sent by a facilitator, and allows the facilitator to join any selected subgroup to listen in and participate if desired. In step 1281, the request is processed first by setting the facilitator as a listener in the selected subgroup, and receives the current speaker data stream or the “no speaker” data stream or “Floor Open” graphic if no one is speaking. The subgroup attendee lists are updated to show that the facilitator is visiting. The facilitator attendee list is also updated. The main loop is then repeated.
[0324] Exit Subgroup
[0325] In step 1282, the request is checked for Exit Subgroup Request. This request is issued by the facilitator when leaving a subgroup, to return to the main conference. In step 1283, the request is processed first to determine if the facilitator was speaking, in which case the “no speaker” data stream or “Floor Open” graphic is displayed to all subgroup participants. The facilitator is removed from the subgroup attendee lists, and the main conference room data stream is received by the facilitator. If there is no speaker in the main conference room, the “no speaker” data stream or “Floor Open” graphic is displayed instead. The facilitator attendee list is also updated, and the main loop is repeated. In a preferred embodiment, the exit subgroup and join subgroup requests can be issued in rapid sequence, or the Join Subgroup request can be used to trigger both processes if the facilitator is currently in a subgroup when the Join request is received. This is to allow the facilitator to go directly from one subgroup to another.
[0326] Reconvene Conference
[0327] In step 1284, the request is checked for Reconvene Conference Request. The request is handled in step 1285 by setting all attendees into the main conference room, updating all of the attendee lists, setting all attendees other than the facilitator as listeners, and transferring control to entry point S8 to set the facilitator as the speaker in the main conference room.
[0328] Unlock Conference
[0329] In step 1286, the request is checked for Unlock Conference Request. In step 1287, the request is processed first by checking the conference mode. If subgroups are active (Multi-Interactive Mode), the request is ignored and the main loop is repeated. Otherwise, in step 1288, the conference lock is turned off, allowing new attendees to join the conference, and the main loop is repeated. In an alternate embodiment, Unlock is allowed, and the conference does not have to be locked in Multi-Interactive Mode. New attendees will appear in the main conference room, and can preferably wait there until the main conference is disbanded, or can be transferred into existing groups by a facilitator.
[0330] Move Subgroup Member
[0331] In step 1290, the request is checked for Move Subgroup Member (Move Attendee) Request. This request is processed beginning with step 1291, which checks to see if subgroups are active. If so, the request is ignored, and the main loop is repeated. If subgroups have not been broken out, in step 1292, the move request is executed, where the selected attendee is moved to the selected subgroup, and the attendee lists are updated, followed by repeating the main loop. In an alternate embodiment, move requests will be honored even if subgroups are active.
[0332] Clear Hands
[0333] In step 1293, the request is checked for Clear Hands Raised Request. If the request check fails, the request has not been recognized, and is ignored by repeating the main loop and discarding the request. The request is processed in step 1294 by resetting the status for all attendees with raised hands in the current group or subgroup, and redisplaying the changes in the attendee lists. Next, the main loop is repeated.

FIGS. 13a and 13b

[0334] FIG. 13a illustrates an attendee interface, with one hand counted.
[0335] FIG. 13a illustrates a recent beta version prototype of the disclosed invention, and is somewhat different from the interfaces in FIGS. 5-9 discussed above. For example, in FIG. 13a, a “hand” icon is used to illustrate a “hand raised” status, whereas a “hand not raised” status is illustrated by a lack of a “hand” icon.
[0336] Referring to FIG. 13b, an alternate embodiment user interface for the present invention, the client application window is displayed. Record Indicator 1301 is displayed, indicating that the conference is being recorded. Meeting Title 1302 displays the title for the meeting set by the organizer, and Time Remaining 1303 shows how many hours, minutes, and seconds remain for the meeting. Clicking on the Settings button 1304, revealing the Settings Panel 1320 described below can access the less used controls.
[0337] The Video Panel 1305 displays either the current speaker’s video (as shown), or a graphic “Floor Open” if no one has the floor, or a graphic “Audio Only” if the current speaker only has a microphone, and no webcam. The current Speaker Name 1307 is displayed at the bottom of the video in a bar where the video or graphic luminance is reduced 50%. Other icons can be displayed in this bar, such as the Hand Raised Indicator 1319 and the Presentation Button 1306 shown. Other items that may be displayed in the bar include a volume indicator (for the speaker only, not for all participants) so the speaker can be aware of the volume of their voice, including a text warning if insufficient volume is detected, such as “Warning: No Audio Signal.”
[0338] The Presentation Button 1306 is presented when the facilitator launches a presentation. This approach requires the participant to click on the button to launch the presentation window. This approach, rather than an automatic launch of the presentation window, is used to avoid issues related to security settings in the browser that has downloaded and
launched the client application. For applications that are not browser-based, direct launch of the presentation window is preferred.

[0339] The Action Button 1308 is the single control button for most functions, including Raise Hand, Lower Hand, Take Floor, and Release Floor (shown). Preferably, this button changes for each possible state, including text, text color, and graphic indicators. In the example shown, the graphic indicators are arrows. Different orientation of the arrows are used for each state: pointing inward for Take Floor, pointed up for Raise Hand, and pointed down for Lower Hand. Other graphic elements may be used in place of the arrows.

[0340] Text communication is carried out via the Text Window 1309 and the Text Entry Area 1310. In this embodiment, the Text Window 1309 scrolls down to reveal the most recent message, unless the participant has scrolled away from the bottom of the text stream to view or capture information previously displayed. Simply clicking within the Text Entry Area 1310, typing a message, and pressing the <Return> or <Enter> key sends a text message to the conference participants. Normal editing functions, such as backspace and arrow keys work as usual. The text entry area may change to a color background to indicate it is active.

[0341] The number of participants in the conference is displayed via the Participant Count 1311, and the status of any vote or poll is displayed via the Voting/Polling Indicator 1312. In this case, a red block, indicating that less than 50% of the available hands are raised, surrounds the raised hand. The number of raised hands is indicated to the left of the indicator. If 50% of the participants have raised their hands, the block is amber, and if more than 50% have raised their hands, the block turns green, thus giving a quick indication of the vote to the facilitator.

[0342] In this embodiment, there are two columns of icons to the left of each participant name in the Participant List 1313. The first column is used for Hand Raised Indicators 1319 and the current Speaker Indicator 1314. The second column of icons indicates the connection type for each participant, such as Video Participant Indicator 1316, Audio Participant Indicator 1317, and Text Participant Indicator 1318. A special icon is used as a Facilitator Indicator 1315. Typically, a facilitator is also a video participant. A raised hand icon may be presented in the video bar as shown to indicate one or more hands are raised. This is necessary because there may be more attendees than can fit in the participant list, and thus raised hands may not be visible in the list unless scrolled to the area where the participant whose hand is raised is located.

[0343] If the participant clicks on the Settings Button 1304, a Settings Panel 1320 is displayed, as shown in FIG. 13c. This panel has various controls, shown here from the facilitator point of view. A non-facilitator participant may see the same panel with control not accessible to them dimmed out, or may see fewer controls. Typically, only a facilitator may use controls such as Bandwidth Controls 1322, Extension Request 1324, Play Intro Video 1325, and Play Presentation 1326. The Misc Controls 1323 allow the participant to enable message alert sounds, enable text chat (facilitator only), and turn on or off the conference recorder (facilitator only). These are described in more detail in conjunction with FIG. 25.

[0344] When the facilitator clicks on the Play Intro Video 1325 button, a video is played in the Video Panel 1305 to all participants. This video may contain a presentation on how to use the conferencing system, designed for new users, and relieves the facilitator from the chore of describing over and over again to new participants how the system works.

[0345] When the facilitator clicks on the Play Presentation 1326 button, the presentation window is launched, and the Presentation Button 1306 is displayed for all participants, allowing them to launch their own window to view the presentation. Presentation controls only appear for facilitators within the presentation window. Additional controls may appear in the Settings Panel, such as access to Flash Settings, Group Formation and Control, and other functions, not shown.

FIG. 14c

FIG. 14a is a first portion of a sign-in diagram. FIG. 14b is a second portion of a sign-in diagram. FIG. 14c is a third portion of a sign-in diagram.

[0347] The sign-in process begins in step 1401 in FIG. 14a. In this process, EC is used to represent an "Event Code" for accessing a specific videoconference. This code is generated by the web application when an event is scheduled, and may be a unique combination of numbers and optionally other characters. For example, a 7-digit number from 0 to 9999999 could be used, allowing up to 10 million unique codes. These codes may be restricted to 1000000 to 9999999 to avoid leading zeros, and may be formatted in various ways, such as NNN-NNNN or NN-NN-NNN, where N represents a numerical digit. For a participant to access a videoconference, he must have the required event code (EC). The process takes place via a web browser that has been sent to the sign-in page of the web application that embodies the sign-in process. Other mechanisms for accessing a particular conference include a unique URL, or a URL with a specific QueryString value. Other ways of identifying a specific conference are well known to one of ordinary skill in the art, and will not be discussed further herein.

[0348] In step 1402, the web server session variable EC is checked for a valid value. If one exists, continue to step 1404 and enter the value in the text entry area automatically for the user. The validity of the session variable is simply that it exists as a string of the required length, or some other similar simple text. For example, checking to see if the format is NNN-NNNN. If no valid EC value exists in the session variable EC, then the user is presented with an empty text entry area in step 1403 to type in the value. Typically, the cursor is automatically moved to the text area input to make it easier for the participant to enter the code.

[0349] In step 1405, the user clicks on a button to check the event code. In this embodiment, the button may be labeled "Check Event." In step 1406, the web application accesses the event database to see if the event code refers to an event that ends in the future. This includes events that have not yet started, as well as events that are ongoing, but does not include events that have already been completed. If no event exists with this code, or the event has already been completed, then an error is reported in step 1407, such as "No Such Event," and the process ends at step 1408. The user may at that point try a different value, starting again at step 1401.

[0350] If the event code matches a videoconference event that ends in the future, the process continues to step 1409, where the web application checks the database to see if this is a paid (Box Office) event. If so, the process proceeds to step B in FIG. 14b. If the process continues to step 1410, where a check is made to see if the participant has logged into an existing account. If not, the process continues to step 1411,
which displays for the user two options: log in or create a new account. Note that it is necessary for a participant to have an account for Box Office events in order to relate a future e-commerce transaction to the account, and thus allow the participant access to the event after paying for it.

[0351] If the participant does not have an account, the process proceeds to step 1412 where the participant sets up an account. Of course, the participant may cancel the process at any time by closing the browser window. Once an account has been set up, the process continues to step 1413 where the participant is logged in automatically. If the participant does have an account in step 1411, the process proceeds to step 1413, where the participant logs into their account, typically by entering a user ID and a password.

[0352] Next, in step 1414, the web application checks the database to see if the participant has already purchased the Box Office ticket for the event. If so, the process continues to step B in FIG. 14b. If not, the user is presented with a shopping cart in step 1415, where the user follows a standard e-commerce checkout procedure in step 1416. This may occur on the web application server, or on a remote e-commerce server. In any case, an e-commerce provider processes the transaction, and appropriate information validating the transaction is posted back to the web application in step 1417. This information includes unique identifiers that allow the web application to access the participant’s account and register the purchase of a Box Office ticket for the specified event.

[0353] Once the e-commerce transaction is completed, the participant may be returned to the sign-in page, as shown to enter the event. In this second pass, the participant is already logged in, and the ticket has been purchased, so the participant is quickly sent on to step B in FIG. 14b.

[0354] Alternatively, the participant closes the web browser, and returns at a later time. This is the case where the participant purchases the ticket several hours or days before the event to facilitate a quick entry to the event when the time arrives.

FIG. 14b

[0355] Referring now to FIG. 14b, step B, the sign-in process continues. In step 1420, the web application checks to see if the event is ongoing (started in the past), or is a future event. If a future event, in step 1422 a check is made to see if the event begins more than 15 minutes in the future. If not a future event, proceed to step 1421, described below. If step 1422 determines that the event starts more than 15 minutes in the future, the process continues to step 1424, where the duration of time between now and the start of the event is displayed, along with a message, such as “Please Check Event again 15 minutes before the start time” for example. Otherwise, in step 1423, the remaining time to when the event starts is displayed, and the process continues to step 1425.

[0356] The value of 15 minutes used here is arbitrary and can be set to any desired value, as long as care is taken to ensure that the participant’s web server session does not expire, or some way is provided to avoid restarting the sign-in process if it does expire, such as the use of cookies or JavaScript® or other mechanisms well known to one of ordinary skill in the art. Typically, a web server session expires in 20 minutes, where all current state for the participant is lost.

[0357] The user may click the Check Event button again at any time—even after the session has expired, since the only value, the Event Code, still resides in the text entry area. However, if the user was logged in prior to the session expiration, the user account information will no longer be available.

[0358] If step 1420 determined that the event starts in the past, the sign-in process 1401 continues in step 1421. Since a validity check has already been made previously in step 1406, we know in step 1421 that the event is ongoing at this time. Thus, the number of minutes that have elapsed since the event began is displayed, such as “event started 25 minutes ago,” for example. The process continues in step 1425.

[0359] Step 1425 is only reached if the event is currently ongoing, or if it starts in 15 minutes or less. Thus, the Event Code is stored in the session variable. This variable is used in the initial sign-in process (steps 1402, 1404) to eliminate the repetitive entry of the Event Code by the participant. Next, in step 1426, the web page is redisplayed, by replacing the welcome message with a message, such as “Click the Event Sign-In button to register,” for example. Also, the Check Event button is redisplayed as an Event Sign-In button, and the Event Code is displayed as text rather than a text entry field. These changes are in addition to displaying any relevant messages, such as “Event began 2 minutes ago,” for example. This change in the page is a convenient way to distinguish between situations where the event is not available and where it is available. Other methods may be used, as well understood by one of ordinary skill in the art, such as displaying a second web page instead of modifying a single web page as described in this embodiment.

[0360] Also, by the use of JavaScript®, the web page may automatically click the Event Sign-In button after a short delay. In this case, the message displayed replacing the welcome message might be “Click the Event Sign-In button to register, or wait for automatic redirect in 5 seconds.” This gives time for the participant to read the message the first time, and allows an experienced participant to immediately click the button without waiting.

[0361] In step 1427, the user clicks the Event Sign-In button, or the JavaScript® clicks the Event Sign-In button after a short delay. In step 1428, JavaScript® code related to the button launches a second page, sized exactly to fit the application window of the video conferencing client application, running on the Flash Player plug-in in other ways well understood by one of ordinary skill in the art, such as a Java® plug-in. This window does not yet contain the video conferencing application, but instead shows the registration window that will be described in relation to FIG. 15 below. The JavaScript® may also force the new window to the front after a short delay, to ensure that it is visible to the participant. This may be necessary if the sized window is smaller than the original sign-in window, and because the sign-in window may return to the front, hiding the new registration window.

[0362] At this point, in step 1429, the background sign-in window is redisplayed in its original form, with welcome message and text, and with the Event Code displayed as already entered in the text entry area. This allows the participant to re-enter the event easily if needed. The process then continues to the registration process described below in relation to FIG. 15.

[0363] In addition to accessing a videoconference event on the web server supporting the main conferencing website, it is also possible to allow remote access to videoconferencing...
within partner websites. One method for doing this is described below, but other methods will be obvious to one of
ordinary skill in the art.

FIG. 14c

[0364] Referring to FIG. 14c, the upper blocks preferably take place on the partner website, and the lower blocks preferably take place on pages not visible to the participant on the main web application site. In step 1430, Remote Registration begins. In step 1431, the partner site does whatever they desire to register a participant. This may include e-commerce, creating an account on the partner’s system, and supplying various pieces of information required by the partner. The process also may include validation steps, such as an email validation that includes the participant receiving an email and clicking on a link to validate their email address.

[0365] In step 1432, the partner web application generates a Registration Code (RC) to uniquely identify this participant. This code could be the account ID for the participant on the partner’s website, for example. Next, in step 1433, a properly sized window is launched via JavaScript®, using a main web application URL provided to the partner. For this example, information is transferred to the main web application via the QueryString associated with the URL. Other methods, such as posting the information to a back-office page on the main web application site may also be used. In this example, the unique Registration Code and the unique Partner Code (PC) is provided.

[0366] As a dotted-line arrow, indicating that this is a transition from activity on the partner site to activity on the main application site. In step 1434, the main web application retrieves the RC and PC values from the QueryString. In step 1435, this information is posted back to the partner site using a provided validation URL at the partner site. This information is provided to the main web application as one of the pieces of information associated with the participant account. In step 1436, the partner site responds with XML data at the validation URL that is read by the main web application. This operation is entirely invisible to the participant. The data can be in other formats, such as text or HTML, for example. The required minimum information being returned is the participant’s name (or “handle”), to be displayed during the videoconference, plus the Event Code.

[0367] In step 1437, the main web application checks to see if the PC and EC values are valid. If so, the process continues in the registration process entry point A in FIG. 15. If the values are not valid, an error message is displayed in step 1438, and the process ends. Validity of the information is typically determined by accessing the database associated with the video conferencing main web application.

FIG. 15

[0368] FIG. 15 is a registration procedure diagram.

[0369] The sign-in process continues to registration in FIG. 15. In step 1502, the participant’s browser cookies are examined to determine if a cookie containing the Attendee ID (AID) exists. This is a unique code assigned to a participant when they have completed the registration process. It is then stored as a cookie on the participant’s computer that expires at the end of the conference so the participant does not need to register more than once, in case of technical difficulties (such as a computer crash, accidentally closing the browser window, etc.).

[0370] If the cookie does not exist, continue to step 1504. If the cookie does exist, in step 1503 a check is made to see if the AID code relates to the Event Code EC. This is determined by looking up the AID code in the participant database, which includes information such as the user name and the Event Code. If a proper match is found, the web application does a bypass of the registration process by going directly to step 1516. If the EC does not match, then the AID cookie is left over from some other event, and the process continues in step 1504.

[0371] In step 1504, the web application retrieves the registration requirements for this event, using the Event Code. Requirements are set by the conference organizer, and may include email, post office address, phone number, or other information. Some of the information may require validation, such as the email address. Validation may also be set as unnecessary by the organizer, as well, during the conference setup process.

[0372] In step 1505, the required blank information fields are presented to the participant. In this embodiment, the participant’s name is always required, but the conference organizer may also require other information. The participant fills in the required information in step 1506 and clicks the Submit button. In step 1507, the information is checked for validity. This may include looking up the information via online databases, such as an address check, or phone number check, or may simply be a check to see if some reasonable content has been provided. For example, if no data is entered in a field, or if the entry is too short to be valid are some simple checks. Phone numbers can be checked against a formula, such as (NNN) NNN-NNNN, for example, where N represents any digit. If the data provided is deemed to be invalid, an error message is displayed in step 1508, and the process returns to step 1506 for data entry.

[0373] Once valid data is received, the process continues to step 1509, where it is determined if an email address is required, and specifically, that it must be verified. It is also possible to not verify the email address (not shown). If no email address was required, the process continues in step 1514. Otherwise, the process continues in step 1510, where the web application checks to see if this participant has logged in. In this case, the participant has an account on the system, and the email can be retrieved by using the validated email in the participant’s account. In an alternate embodiment, the email could have been automatically entered in step 1506, eliminating the need for the participant to re-enter it each time during registration.

[0374] If the participant is logged in, the web application in step 1515 retrieves the participant’s validated email address from their account, and continues to step 1514. If the participant is not logged in, in step 1511 the web application sends an email to the provided email address with a validation link. A validation link can consist of various elements well understood by one of ordinary skill in the art, such as a page URL with additional information in the QueryString that can be used by the referenced page to validate the email address for this participant. In this embodiment, that information might include a unique registration code, for example. The referenced page would then use the registration code to associate the validation process with the specific participant. In an
alternate embodiment (not shown), this registration code could be the proposed AID code to be assigned to the participant in step 1514.

[0375] The web application then presents a message to the participant in step 1512, such as “Validation Email has been sent. Please check both your inbox and your junk mail or spam folder. Click on the link provided to validate your email. Once you have done this, click on the Continue button below.”

[0376] When the participant clicks on the Continue button below, the validation state for the participant is checked in step 1513. If the validation has not been received, the participant is returned to step 1512. If the participant’s email has been validated, the participant is registered in step 1514 using the data provided in step 1506, and a unique AID code is associated with the participant. This unique AID code may be a value used in the validation process as described above, or may be generated as an index into the participant database table. The primary requirement is that it be unique to this participant for this videoconference event.

[0377] Also in step 1514, the assigned AID code is saved on the participant’s machine as a cookie. This process allows rapid re-entry to the event for the participant in the event of some technical difficulties, as described above. Finally, in step 1516, the video conferencing client is launched in the correctly sized window that was launched by JavaScript® in step 1428 of FIG. 14b. Step 1516 may include an additional step or two, such as offering the participant files to be downloaded, if provided by the conference organizer, and an opportunity to set various connection options, such as connection mode (video-audio-text, audio-text, or text-only) and other features, such as use of a proxy server, for example. The final step is to launch the video conferencing client application to allow entry into the conference at the completion of sign-in and registration.

[0378] Step 1514 also represents an entry point from FIG. 14c to step 1437 to complete the process of remote registration and launch of the video conferencing client application.

FIG. 16

[0379] FIG. 16 is a mobile/partner sign-in/registration.

[0380] FIG. 16 depicts an alternate embodiment of a process for mobile device or partner website to sign in and register for a videoconference. The process begins in step 1601. In step 1602, a participant, using either a mobile device or a web browser in an internet-enabled device, enters an Event Code. In alternate embodiments, this code may be provided by the partner website in some manner, or provided automatically as part of processing an invitation. If the mobile device or partner site requires it, a user ID and password may be entered in step 1603.

[0381] The collected information (EC and UID and password, if applicable) is then posted to the main web application in step 1604. Additional information, such as a partner ID may also be passed. This is shown as a gray arrow between steps 1604 and 1605, which indicates this is a passing of information between two separate applications over the Internet.

[0382] In step 1605, the main web application checks to see if the provided EC is valid, i.e., represents a conference that ends in the future. Other checks may be made, such as is the conference related to the partner’s account. This type of check depends on the passing of a partner ID. Also, the ability to operate the system remotely may be a feature that only certain account types have access to, and this type of check can also be made, ensuring that only partners that have the correct type of account may use this remote function. If any of the validity checks fail, the process continues in step 1606, where the web application reports back the error to the partner site or mobile application. This can be done in a number of ways well understood by one of ordinary skill in the art, such as returning a value in XML or in other ways.

[0383] The error code travels over the Internet, as shown in the gray arrow between steps 1606 and 1607, and is received and reported to the partner site or mobile application and to the participant in step 1607. Control then returns to step 1602, where the participant or the partner site or mobile application provides an EC code.

[0384] If the validity checks in step 1605 are successful, a check is then made for a user ID and password in step 1608. If step 1608 determines that no user ID and password are provided, control is passed to step 1613, described below. If step 1608 determines that a user ID and password were provided, the information is then checked for validity in step 1609. If the validity check fails, in step 1610 the partner site or mobile application returns an error to the main web application, in a similar manner as described for step 1606 and 1607. Otherwise, in step 1612, available information is extracted from the participant’s account for registration purposes, such as the participant’s name.

[0385] Next, in step 1613, the event status and any required registration information is returned to the partner site or mobile application. The required registration information is found using EC.

[0386] The partner site or mobile application receives the returned status and registration information over the Internet in step 1614, and requests missing information from the participant in step 1615. One piece of required information is the participant name. The conference organizer, as described above, may require other information. Once the participant has entered the required information, it is then checked as described above. In step 1616, if an email address was requested, the email address is validated in step 1617. This may include sending an email to the participant’s email account with a validation link, as described earlier. If the participant has a verified email address in their account that was accessed using their user ID and password, no additional validation is required.

[0387] Once the email is validated in step 1617, or no email was requested, in step 1618 a determination is made for a Box Office purchase for the conference. If this is a Box Office required ticket purchase, and the participant has not already purchased a ticket, in step 1619 the required e-commerce process is completed.

[0388] Once any Box Office purchase is handled by step 1619, or if the participant has already purchased a ticket, or if this is not a Box Office conference, in step 1620 the registration information collected, plus any required e-commerce information is passed back to the main web application. The registration information typically includes the participant’s name. The e-commerce information, if available, may be stored with the participant’s account to avoid requesting a second payment. Alternatively, the participant purchase information is maintained in the partner site application or mobile application, and is used to determine if a Box Office transaction has already been completed.

[0389] Next, in step 1621, the main web application creates a participant record for the conference, and creates an AID code. Registration information may be written in a database
table associated with the AID code, and may also include the EC. The AID code is then returned to the mobile application or partner site in step 1622, along with an appropriate URL in the main web application that allows the mobile application or partner site to access the videoconference.

[0390] In step 1623, the mobile application or partner site launches the videoconferencing client using EC, AID, and the data access URL. The videoconferencing client then proceeds to connect to the video streaming server (Connect to Server), described below in conjunction with FIG. 17.

FIG. 17

[0391] FIG. 17 is an embedded conference.

[0392] FIG. 17 shows a very similar process as described above for FIG. 16. Specifically, FIG. 17 is a variation of FIG. 16 for an embedded conference or a variation on a mobile application. Rather than repeat the entire process, we will focus on the primary differences between FIG. 16 and FIG. 17.

[0393] The embedded case is where the videoconference actually appears right on a page from the partner site, rather than a separate window launched from the partner site that is tied to the main web application. The same approach described here would also work for a mobile application. For this description, we will focus on the embedded application case, with the understanding that the same process could be used for a mobile application, as well.

[0394] Each of the blocks in FIG. 17 have a related number to the blocks in FIG. 16. For example, block 1704 in FIG. 17 relates directly to block 1604 in FIG. 16. The primary difference between the two figures relates to the collection of and filing of information for registration purposes. Specifically, the embedded application has a relationship with the participant, and not with the videoconferencing provider. Thus, the methods they use to allow a participant into a specific conference is not relevant to the main web application, and no doubt will be handled differently for each partner or application.

[0395] Thus, in steps 1702/1703, information is received for registration purposes. The only required information from the main web application point of view is the participant’s name, the event code, and perhaps the partner or application ID, which can be used to validate that the desired event is actually related to the partner account. This information is gathered by the embedded application in some way, and passed to the main web application in step 1704. Skipping over the description for the main web application steps 1705 to 1713, in steps 1714 and 1715 any information not already provided must be entered. However, it is most likely that all information was provided to complete the registration process for the participant as far as the main web application is concerned right at the beginning (step 1702/1703), and no additional processing is required by the embedded application in step 1714 thru 1720. In effect, this would make it possible to bypass steps 1713 through 1720, and post all required information back to the main web application without additional interaction with the participant. As stated above, the main web application requires only the participant name, EC, and perhaps the partner or application number in order to generate the required AID code and URL to the client application code that will be loaded into the partner website page.

[0396] So in conclusion, FIG. 17 shows a very similar process to a partner or mobile sign-in and registration, but for an embedded case, and that it is highly likely that all registration information is already available at the start, thus allowing step 1704 to post the required information that is then received in step 1721, eliminating all of the intermediate steps.

FIG. 18

[0397] FIG. 18 is a flow diagram of a client connecting to server.

[0398] FIG. 18 is an embodiment of the client connecting to the streaming server. In step 1801 the client makes a connection request with the streaming server passing it the following information: user ID, used as the unique identifier for clients; conference ID, the unique identifier for a conference; user name, used for display purposes in the user interface; facilitator; a variable designating if the client has facilitator privileges in both the client side program and the streaming server; record, used to determine if the conference has the ability to be recorded; hardware (hw) setup, used to determine the device configuration of the client (e.g. microphone, camera). Proceed to step 1802.

[0399] In step 1802 the streaming server receives the information from the client request and determines if this is the first client to connect to the conference. If this is the first client proceed to FIG. 21—Validate Conference. If this is NOT the first client then proceed to step 1803.

[0400] In step 1803 the streaming server checks to see if the conference is valid which is determined in the Validate Conference flowchart. This is set to true when the conference starts so that clients that connect before validation completes are allowed in. If the conference is valid, proceed to step 1804. If the conference is not valid proceed to step 1809.

[0401] In step 1804 the streaming server increases the total user count, used to keep track of the total number of connected clients. Proceed to step 1805.

[0402] In step 1805 all of the connected clients are updated with the information regarding the new user. This is used by the clients to update their UIDs. Proceed to step 1806.

[0403] In step 1806 the streaming server determines if the user ID already exists. Since the system maintains unique user identifiers it is possible for different reasons that a client connects with a user ID that already exists. An example would be if the client opened up the system in one browser and launch the video conferencing in another browser. If the user ID already exists proceed to step 1811. If the user ID does not exist proceed to 1807.

[0404] In step 1807 the streaming server returns a Client Accepted message which informs the client that it is connected to the Video Conferencing System. Proceed to step 1808.

[0405] In step 1808 the client receives the Client Accepted connection. This is the end of the process.

FIG. 19

[0406] FIG. 19 is a flow diagram of a client manage connection.

[0407] FIG. 19 is an embodiment of the client process that manages the connection to the streaming server. The process begins when a successful connection is made with the streaming server (see FIG. 18—Client Connect to Server). Both the client and the streaming server have a process to manage a connection. Every two seconds (preferably) the client sends a check-in message to the server and if the server receives this message it will respond back. If the client does not get a server
response back after 3 attempts (preferably) the connection is reset and if there is a successful reset connection to the server the process begins again. The first step 1901 sets the check_in_tries variable to 0. Proceed to step 1902.

[0408] In step 1902 the client waits 2 seconds to allow time for server to respond to check-in. Proceed to step 1903.

[0409] In step 1903 the client checks to see if check_in_tries is greater than 2. This step is to check if the max number of failed check-ins has reached. If check_in_tries is greater than 2 then proceed to step 1908. If check in tries is less than or equal to 2 then proceed to step 1904.

[0410] In step 1904 the client increments the check_in_tries variable. Proceed to step 1905.

[0411] In step 1905 the client ‘pings’ the streaming server to check in. After sending the ping two steps follow asynchronously. Proceed to step 1906 to continue the streaming servers response of the ping and also proceed to step 1902 where the client waits 2 seconds for the server to respond with an asynchronous callback.

[0412] In step 1906 the server receives the ping and sets the last check_in_time for this client to the current time. Proceed to step 1907.

[0413] In step 1907 the server returns success from the ping call. Proceed to step 1909.

[0414] In step 1908 the client displays a dialog informing the client that it is attempting to reconnect to the streaming server. Proceed to FIG. 18—Client Connect to Server. If the client is successful in that process then the FIG. 19 process begins again.

[0415] In step 1909 the client receives the response back from the server and sets the check_in_tries to 0 which indicates that the client has 3 new attempts to check connection with server.

FIG. 20

[0416] FIG. 20 is a flow diagram of a server manage connection.

[0417] FIG. 20 illustrates an embodiment of the server process that manages the connection to the connected clients. The process begins when the streaming server application launches. Both the client and the streaming server have a process to manage a connection. Every 5 seconds the server checks the last check-in time for all of the connected clients. If any connected client has not checked-in before each 5 second check, that client is disconnected. In step 2001 the server waits 5 seconds before checking the connections. Proceed to step 2002.

[0418] In step 2002 the streaming server retrieves the connected client list. Proceed to step 2003.

[0419] In step 2003 the streaming server gets the first client off of the list. Proceed to step 2004.

[0420] In step 2004 the server checks if the current client has checked in within 5 seconds. If the client has checked in proceed to step 2005. If the client has not check in proceed to step 2007.

[0421] In step 2005 the server checks to see if there are any more clients in the list to check. If there are, proceed to step 2006. If not, proceed back to step 2001.

[0422] In step 2006 the server gets the next client off of the list. Proceed to 2004.

[0423] In step 2007 the server calls the ‘disconnect’ call on the client object which makes sure the client is disconnected from the server perspective as well as removes the client from the server maintained list of connected clients. Proceed to step 2005.

FIG. 21

[0424] FIG. 21 validates a conference.

[0425] An embodiment of the server process that validates the videoconference. This process is executed when the streaming server application is launched, almost always when the first user connects to the system. In step 2101 the Conference_Valid variable is set to ‘YES’ allowing clients to join until the conference has been validated. Proceed to step 2102.

[0426] In step 2102 the streaming server validates the conference with the web server passing it a conference ID. Proceed to step 2102.

[0427] In step 2103 the web server receives the request and determines whether the conference is a valid conference. If it is a valid conference, proceed to step 2104. If it is not a valid conference, proceed to step 2105.

[0428] In step 2104 the web server checks to see if the conference is not only a real conference, but also that the conference start time is less than or equal to the current time AND the conference end time is greater than the current time. If those three criteria are met then the conference is valid, if not then the conference is invalid. Proceed to step 2105.

[0429] In step 2105 the success value is returned to the calling streaming server. Proceed to step 2106.

[0430] In step 2106 the streaming server receives the results and if the conference is valid proceeds to 2107. If the conference is not valid proceed to step 2108. If the response was never received proceed to 2111 which requests the validation again.

[0431] In step 2107 the process ends with the server setting the Conference_Valid to ‘YES’.

[0432] In step 2108 the streaming server disconnects all currently connected users because the conference is NOT valid. Proceed to step 2109.

[0433] In step 2109 the streaming server sets the Conference_Valid to ‘No’ so that all future clients connecting can be refused. This is the end of the process.

[0434] In step 2111 the streaming server waits 3 seconds before making another validation request to the web server.

FIG. 22

[0435] FIG. 22 shows a first and second audio/video stream connection.

[0436] The flow of data and control for a 2-attendee or participant conference is shown in FIG. 22. The upper portion of the FIG. 2201-2204) shows the case where Attendee/participant 1 is speaking and attendee/participant 2 is listening. The lower portion of the FIG. 2211-2214) shows the same configuration, but with the case where attendee/participant 2 is speaking and attendee/participant 1 is listening.

[0437] The base configuration includes the web server 2202, 2212 and the video-streaming server 2201, 2211. Additionally, a database server (not shown) may be included to manage user account data from the web server 2202, 2212. The streaming server and web server may be a single server, although this is not preferred.
[0438] Web server 2202, 2212 provides the browser-based user interface for setting up accounts, paying for accounts, setting up and managing videoconferences, and other housekeeping chores, as described above in relationship to FIG. 3. Typically, account, conference, and attendee/participant information and records are preferably kept in a separate database on the database server (not shown). Web server 2202, 2212 also provides the mechanism for the browser to load the client applications 2203, 2213 and 2204, 2214 that proceed to interact with the web server 2202, 2212 for account and conference information, as well as connecting with and communicating with the streaming server software described in relationship with FIG. 4, running on streaming server 2201, 2211.

[0439] Thus, control information flows between the web server 2202, 2212 and streaming server 2201, 2211 as shown in the diagram, as well as between the client applications 2203, 2204 and both servers 2201, 2211 and 2202, 2212, as shown in the diagram.

[0440] FIG. 22 depicts two audio/video streams between the client application 2203, 2213 and 2204, 2214 and the streaming server 2201, 2211. However, depending on which of the two cases described above, only one of the audio/video streams is active (“Active A/V Stream”) between each client and the streaming server, and one is inactive (“Inactive A/V Stream”), as shown. Thus, audio/video data flows only in one direction—left to right in the upper case (attendee 1 speaking), and right to left in the lower case (attendee 2 speaking). The term “inactive A/V Stream” used herein refers to the possibility of a stream existing, but not an actual stream.

[0441] Thus, control and information flows in both directions, as shown in the control/Status arrows on FIG. 22, while audio/video data only flows unidirectionally, the direction depending on who has the floor.

[0442] Note that if neither attendee/participant has the floor, then all A/V streams are inactive (not shown).

FIG. 23

[0443] FIG. 23 shows a third and fourth audio/video stream connection.

[0444] In FIG. 23, the case for three attendees is shown. The upper diagram (2301-2305) shows attendee/participant 1 speaking, and attendees/participants 2 and 3 listening. The lower diagram shows attendee/participant 2 speaking and attendees/participants 1 and 3 listening.

[0445] As in FIG. 22, control and status information flows in both directions between the streaming server 2301, 2311 and the web server 2302, 2312, as well as between attendee/client applications 2303, 2304, 2305, 2313, 2314, 2315 and the streaming server 2301, 2311 and the web server 2302, 2312.

[0446] However, also as in FIG. 22, A/V streams are unidirectional. In the upper diagram, A/V data flows from attendee/participant 1 2303 and the streaming server 2301, and from the streaming server 2301 and both attendee/participant 2 2304 and attendee/participant 3 2305. The reverse direction A/V streams are inactive.

[0447] The lower diagram in FIG. 23 shows the second case, with A/V data flowing from attendee/participant 2 2314 to streaming server 2311, and from streaming server 2311 to both attendee/participant 1 2313 and attendee/participant 3 2315.

[0448] As with FIG. 22, all A/V streams would be inactive if no one had the floor (was speaking).

[0449] A fifth case (not shown) similarly occurs when Attendee 3 is speaking, and Attendees 1 and 2 are listening.

FIG. 24

[0450] FIG. 24 shows a take floor interface.

[0451] FIG. 24 shows a grey version of a full color embodiment 2400 of the client application 206a or 206b from FIG. 6 of the present inventions, and is similar to FIGS. 13a and 13b with few minor differences. For example, a red flag 2401 is used to indicate a facilitator in addition to the icon change in the next column to the right. This is an alternate way of indicating a facilitator, and can be used in conjunction with a standard video, audio, or text icon in the second column, rather than the special icon shown. Another difference is that no speaker identification bar showing the speaker name 1307 is shown in FIG. 25.

[0452] As in FIGS. 13a and 13b, the interface includes a Settings button 2402. When clicked, a settings panel scrolls down from below the title bar, and is shown in FIG. 25.

FIG. 25

[0453] FIG. 25 is a settings panel.

[0454] The settings panel 2500 is one embodiment of a mechanism to provide additional controls to each participant or facilitator. Other mechanisms are well known by a person of ordinary skill in the art, such as a menu bar, and will not be discussed herein.

[0455] The settings panel 2500 has two views: one for a facilitator (shown), and one for a participant (not shown). The participant settings panel may only include controls that the participant may use, or it may show all controls, with those not available to a participant dimmed out. The advantage of the second approach is to enable a participant to view features available to a facilitator, and thus the second approach acts to some extent as a promotion of features to potential future customers (and as familiarization for future facilitators).

[0456] The settings panel 2500 has several sections, including connection controls 2501, bandwidth selection controls 2502, and other tools 2503. In addition, a button to access Flash Settings 2504 is provided, in case the participant or facilitator is not aware that a right-click will provide access to a menu that includes Flash settings. Also included is an optional Close button 2505, in case the participant or facilitator is not aware that clicking on the Settings button 2402 will both open and close the Settings Panel 2500.

[0457] Connectivity controls 2501 may include a Reset Audio/Video button and a Reset Network button. These buttons are typically seldom if ever needed, and can be eliminated. Bandwidth controls 2502 allow a facilitator to change the bandwidth allowed for the audio/video stream. In this case, we show three values, but any number of values may be provided. The usefulness of this feature is that when one or more participants is connecting via a slow internet connection, or has an older computer that can’t handle a full bandwidth signal, the facilitator may reduce the bandwidth of the conference.

FIG. 26

[0458] FIG. 26 initiates a floor change.

[0459] FIG. 26 is an embodiment of a video conference client initiating a ‘floor change’. In step 2601 the participant clicks on the “Action Button”.

[0460] In step 2602 the state of the floor is checked for the state ‘LISTEN’. This state indicates that the participant does
not have the floor and also does not have a hand raised. If the state of the floor is ‘LISTEN’, then proceed to FIG. 27—Hand Raise/Take Floor, step 2700. If the state of the floor is not ‘LISTEN’, then proceed to step 2603.

[0461] In step 2603 the state of the floor is checked for ‘HAND RAISED’. This state indicates that the client has a “HAND RAISED”. If the current state of the floor is ‘HAND RAISED’, then proceed to FIG. 28—Lower Hand, step 2800. If the state of the floor is not ‘HAND RAISED’, then proceed to FIG. 29—Release Floor which assumes that the client has the floor and is now going to release it.

FIG. 27

[0462] FIG. 27 raises hand and takes floor.

[0463] FIG. 27 illustrates an embodiment of a video conference participant sending a ‘Raise Hand/Take Floor’ request to the streaming media server. In step 2701 the participant (client) sends a ‘Request Floor’ message to the streaming media server.

[0464] In step 2702 the streaming server receives the request and checks to see if the floor is ‘OPEN’, which indicates that no client currently has the floor. If the floor is open, proceed to step 2703. If the floor is not open, which indicates another client has the floor, proceed to step 2709.

[0465] In step 2703 the client the streaming server determines if the client has a microphone and/or camera. If the client has a microphone and/or camera, then the client can stream audio/video and thus is able to have the floor. If this is the case proceed to step 2704. If the participant does not have a microphone or camera, then the participant is only able to send text messages and thus cannot have the floor. If this is the case proceed to step 2709. Preferably a microphone and a camera are both required to be able to take the floor, but a conference of deaf participants may use American Standard Sign Language for “speaking,” and thus would not need necessarily need microphones for taking the floor and “speaking” using sign language.

[0466] In step 2704 the streaming server sets the internal client status to FLOOR and sets the internal Floor Owner to the current client. These variables are used to determine floor ownership. Proceed to step 2705.

[0467] In step 2705 the streaming server updates all connected clients (participants) with the status change of the current client. This is done so that the connected clients can update their UIs. Proceed to step 2706.

[0468] In step 2706 the streaming server sends back a Floor Received response to the client which lets the client know that the client now has the floor. The client can now stream audio/video and make appropriate UI changes.

[0469] In step 2709 the streaming server sets the client’s state to Hand Raised and then in step 2710 increases the total hand raise count. This total is used to inform connected clients of the total number of clients that have hands raised. Proceed to step 2711.

[0470] In step 2711 the streaming server informs all connected clients of the status change of the current client, which is that the client’s hand is raised. Proceed to 2712.

[0471] In step 2712 the streaming server sends a Hand Raised response back to the client to inform the client that the floor is owned by another client, but the requesting client’s hand is raised. Proceed to step 2713.

[0472] In step 2713 the client receives the Hand Raised message from the server and makes the appropriate UI changes.

FIG. 28

[0473] FIG. 28 lowers hand.

[0474] FIG. 28 illustrates an embodiment of a video conference client (participant) sending a ‘Lower Hand’ request to the streaming media server. In step 2801 the client sends a ‘Lower Hand’ message to the streaming media server.

[0475] In step 2802 the streaming server receives the request and checks to see if the client has a hand up. If the client does not have a hand up, the process ends at 2809. If the client does have a hand up, then proceed to step 2803.

[0476] In step 2803 the streaming server sets the client’s state to LISTEN which indicates the client does not have the floor and has no hand raised. Proceed to step 2804.

[0477] In step 2804 the streaming server reduces the total hand raised count. This count is used to keep track of the total number of clients with hands raised. Proceed to step 2805.

[0478] In step 2805 the streaming server updates all connected clients with the new status change of the current client. This update is used to refresh the UIs of the connected clients. Proceed to step 2806.

[0479] In step 2806 the streaming server sends a “Hand Lowered” message to the client. Proceed to step 2807.

[0480] In step 2807 the client receives the “Hand Lowered” message and makes the appropriate UI changes. This ends the process.

FIG. 29

[0481] FIG. 29 releases the floor.

[0482] FIG. 29 illustrates an embodiment of a video conference client (participant) sending a ‘Release Floor’ request to the streaming media server. In step 2901 client preferably stops sending the audio/video stream to server. This is done before the server is notified that the floor has been released because of the buffering that occurs in video streaming. Proceed to step 2902.

[0483] In step 2902 the client waits for 1 second after the audio/video streaming has been stopped. This is related to the video buffering issue above. Proceed to step 2903.

[0484] In step 2903 the client sends a Release Floor message to the streaming server notifying it that the client is no longer streaming audio/video and no longer wants the floor. Proceed to step 2904.

[0485] In step 2904 the streaming server receives the message and determines if the client has the floor. If the client does not have the floor, the process ends. If the client does have the floor, proceed to step 2905.

[0486] In step 2905 the Floor Owner, an internal server variable, is cleared to show that no client has the floor.

[0487] In step 2906 the streaming server sets the client’s state to Listen, meaning the client no longer has the floor. Proceed to step 2907.

[0488] In step 2907 the streaming server updates all clients regarding the status of the requesting client informing them that the client no longer has the floor. Proceed to FIG. 30a—Next Speaker which will select a client with a hand raised and then proceed to step 2908.

[0489] In step 2908 the streaming server sends a Floor Released message to the client informing it of its new floor state. Proceed to step 2909.
In step 2909 the clients receive the Floor Released message and makes the appropriate UI changes. The release floor process ends here.

FIGS. 30a AND 30b

FIG. 30a shows a first portion of going to the next speaker. FIG. 30b shows a second portion of going to the next speaker. FIGS. 30a and 30b are an embodiment of the process on the streaming server to find the next client (participant) with a hand raised and give that client the floor. In step 3001 the server retrieves the client list which is maintained by the streaming server. Proceed to step 3002.

In step 3002 the streaming server retrieves the first client name from the list. A client on the list includes variables that are native to the streaming server, but also variables that can be added unique to the application (e.g. floor state, hardware setup, and most importantly for our purposes here, time when hand was raised). Proceed to step 3003.

In step 3003 the streaming server determines if the current client’s floor state is Hand Raised. If the current client has its hand raised, then proceed to 3004. If the client does not have a hand raised, then proceed to step 3006 where the server checks to see if there are any more clients in the list.

In step 3004 the server determines if this is the first raised hand found in this list. If it is, proceed to step 3008. If it is not the first raised hand found, proceed to step 3005.

In step 3005 the streaming servers sends a Floor Open message to all connected clients informing them that the floor is now able to stream audio/video and make the appropriate UI changes. This is the end of the process.

In step 3006 the streaming server sends a Floor Open message to all connected clients informing them that the floor can now be taken by any interested clients. This is the end of the process.

FIGS. 30a AND 30b

[0491] In step 2909 the clients receive the Floor Released message and makes the appropriate UI changes. The release floor process ends here.

[0492] FIGS. 30a and 30b are an embodiment of the process on the streaming server to find the next client (participant) with a hand raised and give that client the floor. In step 3001 the server retrieves the client list which is maintained by the streaming server. Proceed to step 3002.

[0493] In step 3002 the streaming server retrieves the first client name from the list. A client on the list includes variables that are native to the streaming server, but also variables that can be added unique to the application (e.g. floor state, hardware setup, and most importantly for our purposes here, time when hand was raised). Proceed to step 3003.

[0494] In step 3003 the streaming server determines if the current client’s floor state is Hand Raised. If the current client has its hand raised, then proceed to 3004. If the client does not have a hand raised, then proceed to step 3006 where the server checks to see if there are any more clients in the list.

[0495] In step 3004 the server determines if this is the first raised hand found in this list. If it is, proceed to step 3008. If it is not the first raised hand found, proceed to step 3005.

[0496] In step 3005 the streaming servers sends a Floor Open message to all connected clients informing them that the floor is now able to stream audio/video and make the appropriate UI changes. This is the end of the process.

[0507] In one embodiment, the conference account owner has additional options to control access to a scheduled conference. These additional options may be associated with various different account levels, based on pricing or some other factor. One additional option is called Open Seating. This option allows additional participants into the conference beyond the number set by the conference owner. The acceptance of additional participants using this option may depend on available conference center seating as well as available funds in the owner’s account.

[0508] Another option is called PrePay Boost. This feature is designed to allow the owner to expand the number of seats in a scheduled conference beyond the maximum number contracted for, based on available funds in his account and conference center seating. This differs from Open Seating because seats are guaranteed, rather than being dependent on available seating, such as in the case where the conference center is full from many other ongoing conferences. The advantage of Open Seating, however, is that the owner only has to pay for seats actually occupied. For PrePay Boost, the seats are paid for and guaranteed even if not used in the conference.

[0509] Another option is called Box Office. This feature allows the owner to set a price for attending the conference. The price is extracted from the participant using e-commerce in a manner that is well understood by one of ordinary skill in the art. The received funds or a portion thereof may be deposited in the owners account. A percentage of the funds may be taken out for the service provided.

[0511] Another option is Registration Options. Here, the owner may specify what information he requires from participants, such as email, phone, or address. The email address may be validated before allowing the participant into the conference by sending a validation email, where the participant must click on a link to validate the email. Phone and address may also be validated, based on the participant’s name, which also may be required. This feature allows the conference owner to collect verified information about the participants for other uses, such as marketing.
controls. Other controls may be included for both facilitators and non-facilitators, such as window sizing controls and window closing controls.

[0512] Another option for a facilitator may have is the ability to modify the bandwidth of the video/audio streams for all participants. This is useful if one or more of the participants have insufficient Internet bandwidth to handle the video/audio stream. This option is controlled by a control, such as a radio selector in the settings panel.

[0513] Another option is called Secure Meeting. This option is useful for meetings where there is concern of someone intercepting the video streams and recording or listening to the conversation. If this option is selected, all audio/video streams are encrypted using mechanisms that are well understood by one of ordinary skill in the art.

Summary

[0514] The present invention has been described above with reference to a preferred embodiment. Much of the description centers around the use of a keyboard and mouse, but other embodiments, such as touch-screen-based tablets and smartphones are also envisioned, where gestures or touches are used rather than mouse clicks and typing. These and other embodiments will be apparent to those skilled in the art in light of this disclosure. For example, the present invention may readily be implemented using configurations other than those described in the preferred embodiment above, such as using fixed hard-wired connections via a switched network rather than the Internet. Additionally, the present invention may effectively be used in conjunction with systems other than the one described above as the preferred embodiment. Therefore, those and other variations upon the preferred embodiments are intended to be covered by the present invention, which is limited only by the appended claims.

[0515] While the foregoing has described what are considered to be the best mode and/or other examples, it is understood that various modifications may be made therein and that the subject matter disclosed herein may be implemented in various forms and examples, and that the teachings may be applied in numerous applications, only some of which have been described herein. It is intended by the following claims to claim any and all applications, modifications and variations that fall within the true scope of the present teachings.

What is claimed is:

1. A video teleconferencing method for routing video streams in a system, the system including a server, a first device of a first attendee, and a second device of a second attendee, the method comprising:
   - creating a first state by selecting, by the server, the first attendee as the speaker;
   - receiving, by the server during the first state, the first video stream from the first device;
   - sending, by the server during the first state, the received first video stream to the first device;
   - inactivating, by the server during the first state, the first video stream from the server to the first device;
   - inactivating, by the server during the second state, the first video stream from the server to the second device;

2. The video teleconferencing method of claim 1, further comprising:
   - creating a second state by:
     - deselecting, by the server, the first attendee as the speaker;
     - receiving, by the server during the second state, the second video stream from the second device;
     - sending, by the server during the second state, the received second video stream to the first device;
     - inactivating, by the server during the second state, the first video stream from the first device to the server;
     - inactivating, by the server during the second state, the first video stream from the server to the second device;

3. A video teleconferencing method for routing video streams in a system, the system including a server, a first device of a first attendee, a second device of a second attendee, and a third device of a third attendee, the method comprising:
   - creating a first state by selecting, by the server, the first attendee as the speaker;
   - receiving, by the server during the first state, a first video stream from the first device;
   - sending, by the server during the first state, the received first video stream to the second device;
   - sending, by the server during the first state, the received first video stream to the third device;
   - inactivating, by the server during the first state, a second video stream from the second device to the server;
   - inactivating, by the server during the first state, the second video stream from the server to the first device;
   - inactivating, by the server during the first state, a third video stream from the third device to the server;
   - inactivating, by the server during the first state, the third video stream from the server to the second device;
   - inactivating, by the server during the first state, the third video stream from the server to the third device;

4. The video teleconferencing method of claim 3, the method further comprising:
   - deselecting, by the server, the first attendee as the speaker, and
   - selecting, by the server, the second attendee as the speaker;
   - receiving, by the server during the second state, the second video stream from the second device;
   - sending, by the server during the second state, the received second video stream to the first device;
   - inactivating, by the server during the second state, the first video stream from the first device to the server;
   - inactivating, by the server during the second state, the first video stream from the server to the second device;
   - inactivating, by the server during the second state, the third video stream from the third device to the server;

5. The video teleconferencing method of claim 4, the method further comprising:
creating a third state by:
  deselecting, by the server, the second attendee as the speaker; and
  selecting, by the server, the third attendee as the speaker;
  receiving, by the server during the third state, the third video stream from the third device;
  sending, by the server during the third state, the received third video stream to the first device;
  sending, by the server during the third state, the received third video stream to the second device;
  inactivating, by the server during the third state, the first video stream from the first device to the server;
  inactivating, by the server during the third state, the first video stream from the server to the second device;
  inactivating, by the server during the third state, the first video stream from the server to the third device;
  inactivating, by the server during the third state, the second video stream from the second device to the server;
  inactivating, by the server during the third state, the second video stream from the server to the first device; and
  inactivating, by the server during the third state, the second video stream from the server to the second device.

6. A video teleconferencing method for controlling video streams in a system, the system including a server, a plurality of attendees each respectively having an associated communication device, the method comprising:
  allowing, by the server, only one speaker to exist at a time at a teleconference, wherein the speaker is one of the attendees;
  sending, from the speaker’s device, a speaker’s video stream to the server;
  receiving, by the server, the speaker’s video stream;
  sending, by the server, at least a portion of the speaker’s video stream to at least one non-speaker attendee; and
  preventing any video stream from any non-speaker attendee from being sent through the server to other attendees.

7. The video conferencing method of claim 6, the method further comprising:
  ordering the devices of all non-speaker attendees to refrain from sending video streams to the server.

8. The video conferencing method of claim 6, the method further comprising:
  permitting only the speaker’s device to send the speaker’s video stream to the server.

9. The video conferencing method of claim 6, the method further comprising:
  ordering the devices of all non-speaker attendees to refrain from sending video streams to the server; and
  permitting only the speaker’s device to send the speaker’s video stream to the server.

10. The method of claim 6, wherein the speaker has the power to withdraw from being the speaker, and wherein the speaker must withdraw from being the speaker before new speaker can exist.

11. The method of claim 6, wherein, after an initialization, only the speaker can select a new speaker.

12. The method of claim 6, wherein a current speaker is limited to being the current speaker for a predetermined period of time, after which only a non-speaker assignee can become a new speaker.

13. The method of claim 6, wherein the speaker has the power to withdraw from being the speaker, and withdrawal allows non-speakers to become speakers by taking a floor.

14. The method of claim 6, wherein at least one of the attendees is a facilitator attendee, the method further comprising:
  providing the facilitator with a power to select a new speaker.

15. The method of claim 6, further comprising:
  providing the facilitator with a power to remove a current speaker to open a floor for a new speaker.

16. The method of claim 6, further providing at least one non-facilitator attendee with a process for requesting to become the new speaker.

17. The method of claim 16, wherein the request is transmitted to the facilitator.

18. The method of claim 16, wherein the request is transmitted to the speaker, and wherein the speaker has the power to select the requesting attendee to become the new speaker.

19. The method of claim 13, further comprising:
  providing a second facilitator with the power to select the new speaker.

20. The method of claim 13, further comprising:
  providing the facilitator with the power to create and control subgroups.

21. The method of claim 20, further comprising:
  allocating all of the non-facilitator attendees among the subgroups.

22. The method of claim 20, wherein the allocation is performed by the facilitator.

23. The method of claim 20, wherein the allocation is performed automatically according to a predetermined rule.

24. The method of claim 20, wherein the allocation is performed automatically according to a rule selected by the facilitator.

25. The method of claim 24, wherein a manual reallocation is performed by the facilitator after the automatic allocation.

* * * * *